Rapid solution of potential integral equations in complicated 3-dimensional geometries

by

Joel Reuben Phillips

S. B., Massachusetts Institute of Technology (1991)
S. M., Massachusetts Institute of Technology (1993)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1997

© 1997 Massachusetts Institute of Technology

All rights reserved.

19980115 185

Signature of Author . . . .
Department of Electrical Engineering and Computer Science
June 6, 1997

Certified by . . . .
Jacob K. White
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

APR 2 8 1997

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
June 1997

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Rapid solution of potential integral equations in complicated 3-dimensional geometries

5. FUNDING NUMBERS

6. AUTHORS
Joel Reuben Phillips

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology

AFRL-SR-BL-TR-98-
0016

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NI
110 Duncan Avenue, Room B-115
Bolling Air Force Base, DC 20332-8080

AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Analysis of many electromagnetic problems in engineering, such as electromagnetic interference (EMI) calculations or estimation of interconnected coupling capacitances and inductance's, is often performed via numerical methods based on integral equations. Analysis of the complicated three-dimensional geometries of modern engineering structures required efficient algorithms to solve the large, dense linear systems generated by integral equation techniques. This thesis develops and analyzes a grid-based, "precorrected-FFT" method which preserves the efficiency of recently developed fast-multipole techniques but is more easily generalizable to a variety of kernels, and may have substantial performance benefits for commonly encountered geometries. The proposed algorithm is intended to be particularly efficient for problems with Helmholtz kernels where the size of the problem domain is of the order of a few wavelengths, such as typically occurs in EMI calculations, and for problems with planar interfaces, such as integrated circuits or interconnect in layered media (dielectrics and groundplanes).

DTIC QUALITY INSPECTED 3

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239.18
Designed using WordPerfect 6.1, AFOSR/XPP, Oct 96

Rapid solution of potential integral equations in complicated 3-dimensional geometries

by

Joel Reuben Phillips

Submitted to the Department of Electrical Engineering and Computer Science
on June 6, 1997, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Analysis of many electromagnetic problems in engineering, such as electromagnetic interference (EMI) calculations or estimation of interconnnect coupling capacitances and inductances, is often performed via numerical methods based on integral equations. Analysis of the complicated three-dimensional geometries of modern engineering structures requires efficient algorithms to solve the large, dense linear systems generated by integral equation techniques. This thesis develops and analyzes a grid-based, "precorrected-FFT" method which preserves the efficiency of recently developed fast-multipole techniques but is more easily generalizable to a variety of kernels, and may have substantial performance benefits for commonly encountered geometries. The proposed algorithm is intended to be particularly efficient for problems with Helmholtz kernels where the size of the problem domain is of the order of a few wavelengths, such as typically occurs in EMI calculations, and for problems with planar interfaces, such as integrated circuits or interconnect in layered media (dielectrics and groundplanes).

Thesis Supervisor: Jacob K. White
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

First and primary thanks must go to Professor Jacob White, who has been a constant source of advice and support for much of my tenure at MIT. Without him this thesis could not have come to be.

I would also like to thank Professors Steve Senturia and Terry Orlando, who were kind enough to act as readers for the thesis. Both provided guidance and encouragement throughout my career.

Matt Kamon has been a patient officemate and foil for spontaneous technical discussions.

I would like to thank the staff of the IBM Thomas J. Watson Research Center for their hospitality. Much of the work for Chapters 8 and 9 was performed there. In particular, Eli Chiprout, David Ling, and Albert Ruehli contributed greatly to this work.

To all the past and current members of the 8th floor, and the myriad others who have provided technical and emotional support over many years, thank you for your friendship. You know who you are.

The work needed to produce this thesis was also supported in part by DARPA Contracts DABT63-94-C-0053 and J-FBI-92-196, the Consortium for Superconducting Electronics, SRC under contract SJ-558, NSF, an NDSEG fellowship, and an IBM fellowship.

One cannot think well, love well, sleep well, if one has not dined well.

— Virginia Woolf

Contents

Abstract 3

Acknowledgments 5

List of Figures 11

List of Tables 16

1 Introduction 19

2 Approximation of potential functions 25

2.1 The Approximation Problem 25
2.2 The Grid-Potential-Collocation Approach 28
2.3 Error analysis 30
2.4 Helmholtz Kernels 34
2.5 Utilizing the far-field expressions 38
2.6 Applications and competing approaches 38
2.7 Empirical Grid Error Analysis 39

3 The Precorrected-FFT Algorithm 49

3.1 The Basic Idea 49
3.2 Projecting onto a grid 51
3.3 Computing Grid Potentials 52
3.4 Interpolating Grid Potentials 52
3.5 Precorrecting 53
3.6 Complete Algorithm 54
3.7 Grid selection 55

4 Complexity Analysis 57

5 Algorithm Performance on Laplace Problems **63**
5.1 Empirical Error Analysis 64
5.2 Reference Examples 64
5.3 Realistic Examples 71

6 The Laplace Equation in Stratified Media **77**
6.1 Motivation 77
6.2 Calculating Green functions for planar geometries 79
6.2.1 Green function derivation 79
6.2.2 Approximating the Green functions 81
6.3 Algorithmic Implications 82
6.3.1 The interpolation/projection operators 82
6.3.2 Modifications to the FFT-based convolution 83
6.3.3 Constraints on grid size and spacing. 86
6.4 Examples 87

7 Helmholtz Problems **95**
7.1 Formulation and Discretization 95
7.2 Computational Examples 99

8 Accelerated full-wave EM analysis **107**
8.1 The rPEEC formulation 108
8.2 Grid-based matrix sparsification 109

9 Model Reduction Issues **113**
9.1 Rational approximation via orthogonalized Krylov methods 114
9.2 Reduction of distributed systems 116
9.3 Multipoint rational approximants 118
9.4 Computational Examples 119
9.5 Obtaining time-domain models 121
9.6 Difficulties of the present approach 126

10 Conclusion **129**

Bibliography **131**

List of Figures

1-1 Piecewise-constant collocation discretization of two conductors. Conductor surfaces are discretized into panels which support a constant charge density. 20

1-2 Computational resources needed by a 100MFLOPS workstation to perform dense LU factorization. 21

2-1 Left: Interpolation of the $1/r$ Green function. Interpolation is accurate away from the origin but inaccurate near the singularity. Right: Polar plot of the magnitude of the potential of a charge distribution far away from the sources. . 26

2-2 Potentials $\psi(r < r_c, \theta, \phi)$ or $\psi(\tilde{r} > r_c, \tilde{\theta}, \tilde{\phi})$ for $r, \tilde{r} > a$ may be obtained from the potential at $r = r_c$. 29

2-3 2-D Pictorial representation of the grid projection scheme. The black points (at $\hat{x}$) represent the grid charges ($\hat{q}$) being used to represent the triangular panel's charge density σ. The white points are the the points x^t where the potential due to the black point charges and the potential due to the triangular panel's charge density are forced to match. The grid charges approximate the panel potential outside the gray region. 30

2-4 Left: Interpolation of the real part of the e^{ikr}/r Green function. Interpolation is inaccurate near the singularity as in the $1/r$ case, but may also be inaccurate away from the singularity if the spacing of the interpolating points is large compared to the characteristic wavelength $2\pi/k$. Right: Polar plot of the magnitude of the potential of a hypothetical charge distribution in the far-field limit. 35

2-5 Error vs. effective order for multipole expansions (x) and grid-projection scheme (*), at first-near-neighbors cells. For an order-M quadrature rule, the effective order of the grid-projection scheme is $(M - 1)/2$. 40

2-6 Effective number of coefficients per cell vs. effective order for multipole expansions (x) and grid-projection scheme (*). 41

2-7 Error vs. effective number of coefficients per cell for multipole expansions (x) and grid-projection scheme (*), at first-near-neighbor cells. 41

2-8 Error in grid approximations. From bottom to top, $p = 6, 5, 4, 3, 2$. 43

2-9 Error in grid approximations. From bottom to top, $p = 6, 5, 4, 3, 2$. 44
2-10 Error in grid approximations. From bottom to top, $p = 6, 5, 4, 3, 2$. 45
2-11 Error in grid approximations. From bottom to top, $p = 6, 5, 4, 3, 2$. 46

3-1 (a) Side view of a sphere discretized into 320 panels, with spatial decomposition into a $3 \times 3 \times 3$ array of cells. (b) Superimposed grid charges corresponding to the cell decomposition of (a), with $p = 3$. In each cell, a $3 \times 3 \times 3$ array of grid charges is used to represent the long range potential of the charged panels in the cell. Some of the grid charges are shared among cells. Note that the grid is "coarser" than the triangular panels used to discretize the sphere. The grid extends outside the problem domain because the number of grid points is required to be a factor of two. 50
3-2 2-D Pictorial representation of the four steps of the precorrected-FFT algorithm. Interactions with nearby panels (in the grey area) are computed directly, interactions between distant panels are computed using the grid. 50

5-1 (a) A sphere discretized into 960 panels. The discretization is refined by subdividing the spherical triangle defined by the panel vertices into four triangular panels, whose vertices are the midpoints of the edges of the original spherical triangle. (b) Solid lines show integrated charge error for the sphere with Dirichlet condition of Eq. 17. Solid line shows errors for grid code, (x) $p = 2$ (*) $p = 3$ (+) $p = 4$. Dashed line connecting (o) shows error for $l = 2$ multipole scheme. . 65
5-2 The cube example. (a) Discretization of the cube. (b) CPU time, in seconds, needed for the fast multipole (dashed line connecting 'x') and precorrected-FFT algorithms (*) to compute a matrix-vector product. For the precorrected-FFT algorithm, different results are possible depending on whether speed or memory usage is to be optimized. The solid line connects runs with grid sizes chosen to minimize memory use. (c) Memory, in Mb, needed by the fast multipole and precorrected-FFT algorithms. (d) Product of (b) and (c). Note the speed-memory product is fairly independent of grid size. 66
5-3 The nonuniformly-discretized cube example. 86,400 panels were present in the discretization shown. (b)-(d): Computational resources needed for the fast multipole (x) and precorrected-FFT algorithms (o) to compute a matrix-vector product for the nonuniformly discretized cube geometry. (*) shows time needed by the precorrected-FFT algorithm for the uniform-cube discretization. (b) CPU time, in seconds, (c) Memory, in Mb, needed by the fast multipole and precorrected-FFT algorithms. (d) Product of (b) and (c). 68

5-4 The bus crossing example. (a) Larger problems are generated by adding more bus lines. (b) CPU time, in seconds, needed for the fast multipole (dashed line connecting 'x') and precorrected-FFT algorithms (solid line connecting '*') to compute a matrix-vector product. (c) Memory, in Mb, needed by the fast multipole and precorrected-FFT algorithms. (d) Product of (b) and (c). 70

5-5 Several realistic capacitance extraction problems. (a) The woven bus example (woven5x5). (b) the comb drive example (comb). (c) The via example (via). (d) the SRAM example(SRAM). 73

5-6 The large woven bus structure. The 30 conductor structure is formed from fifteen conductors woven around fifteen straight conductors. The actual discretization is finer than shown in the above figure. There are 82,080 panels in the actual discretization. 75

6-1 Two-interface layered dielectric problem. Interfaces extend to infinity in x and y directions. 80

6-2 The grid spacing difficulty. If the grids in two layers have unequal spacing, $d_{32} \neq d_{21}$ and the grid-grid mapping matrices H_{mn} will not have Toeplitz structure. . 86

6-3 The two sphere problem with discretized interface. Actual discretization is shown. 89

6-4 The two hemisphere problem with discretized interface. Actual discretization is finer than shown. 91

6-5 The airbridge example. Actual discretization is finer than shown. 93

7-1 Condition number κ of matrix P vs. wavenumber k for 240-panel sphere, derived from integral equation based on single-layer (SL, $\eta \rightarrow \infty$), double-layer (DL, $\eta = 0$), and combined-layer(CL, $\eta = 1$) potentials. 96

7-2 Effect of choice of integral equation on numerically computed scattered fields near interior resonances. Solid line: real part of exact solution. Dashed line: imaginary part of exact solution. x: computed real part of solution. +: computed imaginary part of solution. Top plots show single-layer potential, bottom plots show combined-layer potentials, at $ka = 3.0$ and $ka = 3.2$. The single-layer solution is inaccurate around the resonance $ka = \pi$, but accurate elsewhere. The combined-layer solution is accurate near both away from the resonance and near it. No matrix approximations were performed for these computations. 97

7-3 The 3840 panel discretization of the sphere. 99

7-4 Errors in relative charge density for plane-wave excitation, vs. wavenumber and $1/n$, $n =$ number of panels in discretization. Solid line: $p = 3$, dash line: $p = 4$, dash-dot line: $p = 2$. 100

7-5 Scattered fields for plane-wave boundary condition on sphere, $ka = 25$. Solid line: real part of exact solution. Dashed line: imaginary part of exact solution. x: computed real part of solution. +: computed imaginary part of solution. . . 101
7-6 Computational resources needed to solve the combined-layer potential integral equation on sphere of radius $a = 1$. (x) Fixed-k experiment. (*) Variable-k experiment. (–) Linefit to variable-k points. (-.) Linefit to $k = 0.5$ points. . . . 102
7-7 (o) Resources required for FMM, $l = 2$, to solve Eq. 1.2 . (*) variable-k experiment from Fig. 7-6 Left figure shows CPU time needed for matrix-vector product, right figure shows needed memory, in MB. 103
7-8 Product of cpu time-memory for matrix-vector products for 1% error solution of Helmholtz equation on sphere.(o) polynomial interpolation operators, (x) grid-collocation operators. 105

9-1 Frequency response of the two-strip example. Left (a): Amplitude of driven current vs. excitation frequency, 100 MHz to 3 GHz. Solid lines show directly computed response and Arnoldi-based approximation of dense model. Dashed line shows AWE approximation of dense model. 'x' shows response at selected frequency points of sparse model, computed using the iterative algorithm GMRES. Right (b): maximum amplitude of electric field at 3 meters from structure. Solid line shows reduced-order model of dense system. Dashed line shows reduced-order model of approximate sparse system. 'x' shows full dense model. . 119
9-2 Left : a large interconnect structure. Vertical axis is not to scale. Right: Current driven by voltage source through resistive termination to ground plane. Dotted line: order-20 Arnoldi model with $s_0/(2\pi) = 0.5 + 2i$ GHz. Dash line: order-15 Arnoldi model with $s_0/(2\pi) = 0.5$ GHz. Solid line: response obtained without model reduction, as well as the piecewise-rational approximation obtained from combining models at separate expansion points; the results overlap. Note that the expansion at $0.5 + 2i$ Ghz gives a very good approximation up to 3 GHz but is not good near $s = 0$ whearas the opposite is true of the expansion at 0.5 GHz. 120
9-3 (x) Poles calculated by Arnoldi method. (*) poles calculated by moment-matching. The right hand plot is an expanded view of the poles near the origin. 121
9-4 Frequency response functions for 30-cm long strips with 10-Ω termination driven by voltage source with 50-Ω internal impendance. (a) Magnitude of $H(s) = I(s)/V(s)$. Solid line shows overlapped multipoint frequency response, (+) show frequency response of final model after deletion of unstable poles and recalculation of residues. (b) phase of $H(s)$, same figure key as for (a). (c) Amplitude of $H(s)$, expansion point $s_0 = 0.5 + 0i$ Ghz (d) Amplitude of $H(s)$, expansion point $s_0 = 0.5 + 2i$ Ghz (e) Amplitude of $H(s)$, expansion point $s_0 = 0.5 + 4i$ Ghz (f) Amplitude of $H(s)$, expansion point $s_0 = 0.5 + 6i$ Ghz 123

9-5 Time-domain step response of 30-cm long strips. 124
9-6 Time-domain step response of 30-cm long strips with 10-Ω termination, varying input rise times t_r. 125

List of Tables

2-1 Quadrature rules used for the grid-projection scheme experiments. 42

2-2 Maximum errors at first- and second-near-neighbor cubes for various grid-projection schemes. 47

5-1 Statistics for FASTCAP Order-2, Grid-2,3 codes for $1/r$ Green function. Setup, solve, and CPU times are in seconds on DEC AXP 3000/900, memory in megabytes. m is number of conductors in problem, each conductor requires a separate linear system solution. 74

5-2 Comparison of FASTCAP and grid codes. Figures are ratios of required resources. "Product" is product of CPU and memory figure. 74

5-3 Comparison of errors in capacitances induced by using grid-collocation operators for interpolation and polynomials for interpolation. Both columns use grid-collocation for the grid projection step. 75

5-4 Comparision of capacitance extraction algorithms. Figures in parentheses are estimates. 76

6-1 Statistics for FFTCAP-layered media code, FFTCAP free-space code, and FASTCAP. Setup, solve, and CPU times are in seconds on DEC AXP 3000/900, memory in megabytes. m is number of conductors in problem, each conductor requires a separate linear system solution. The LM code used the ground-plane Green function, the FS codes use the free-space Green function, and have no equivalent-charge discretized-interfaces for these problems. 87

6-2 Comparison of layered-media code to free-space FFTCAP and FASTCAP codes for the problems of Table 6-1 . Figures are ratios of resources required by the layered media code to those needed by the free-space precorrected-FFT code (FFT in table) or the free-space FASTCAP (FMM) code. 88

6-3 Two sphere example. Table gives computational resources required by the layered media precorrected-FFT code [FFT-LM] and the free-space precorrected-FFT [FFT-FS] and fast-multipole [FMM] codes. The free space codes use discretized interfaces to account for the planar portion of the geometry, except for the symmetry plane case, which is implemented by duplication of the conductor discretiztion. 89

6-4 Mutual (C_M) and self (C_S) capacitance, in pF, of spheres in dielectric half-spaces, with varying dielectric constants. Spheres are of radius 1 with centers located 1.05 above interface. X indicates the non-preconditioned GMRES solver did not converge in available memory. Superscript "DI" indicates capacitances calculated using the equivalent charge formulation with discretized interfaces. . . 90

6-5 Mutual (C_M) and self (C_S) capacitance, in pF, of hemispheres in dielectric half-spaces, with varying dielectric constants. Superscript "DI" indicates capacitances calculated using the equivalent charge formulation with discretized interfaces. X indicates the non-preconditioned GMRES solver did not converge in available memory. 92

6-6 Two hemisphere example. Table gives computational resources required by the layered media precorrected-FFT code [FFT-LM] and the free-space precorrected-FFT [FFT-FS] and fast-multipole [FMM] codes. The free space codes use discretized interfaces. 92

6-7 Realistic and multi-interface examples. m is the number of non-interface conductors in the problem, n the number of panels, excluding the interface. 94

7-1 Resources required to solve scalar Helmholtz equation on a sphere of diameter d discretized into 61440 panels. Memory and times are normalized to those required for the static precorrected-FFT code. Parameters chosen for 1% relative error in charge density. More iterations are needed for $ka = 0$ than $ka = 0.5$ because the $ka = 0$ codes use the less well-conditioned single-layer formulation. 105

1

Introduction

The use of discretized integral equations for engineering analysis of electromagnetic problems has been popular at least since the appearance of Harrington's treatise[1] nearly thirty years ago. Algorithms using method of moments [1] or weighted-residuals [2, 3] based discretizations of integral equation formulations are often attractive for problems with complicated problem domains because of the reduced complexity of discretization when compared with competing approaches such as finite-element methods. For example, boundary-element methods[4] for solving elliptic partial differential equations require discretization only of domain boundaries and not of interior or, particularly, exterior volumes. The difficulty with such approaches is that they generate dense matrix problems which are computationally expensive to solve, and this limits the complexity of problems which can be analyzed. Recently, the combination of iterative linear system solution algorithms and matrix "sparsification" techniques has been used to create efficient integral-equation codes [5, 6, 7] and has resulted in renewed interest in integral-equation methods.

To illustrate the difficulties associated with discretized integral equations, consider the model problem of capacitance extraction. The capacitance of an m-conductor geometry is given by a (symmetric) matrix $C \in \mathbf{R}^{m \times m}$. The entry C_{kl} represents capacitive coupling between conductors l and k. In a general sense, the capacitance matrix may be determined from the matrix equation

$$CV = Q \tag{1.1}$$

where the columns of $V \in \mathbf{R}^{m \times m}$ are linearly independent "test" vectors. V_{kl} represents the potential of the kth conductor, for the lth test vector. $Q \in \mathbf{R}^{m \times m}$ is a matrix of conductor charges, that is, Q_{kl} is the charge on conductor k at the lth test. For example, suppose the potential of each conductor l is set to unity in turn and the potential of the remaining conductors set to zero (this corresponds to the choice of the lth column of V to be the lth unit vector). The resulting conductor charges (the lth column of Q) will give the lth column of the capacitance matrix, i.e., the charge Q_{kl} on conductor k is the capacitance C_{kl}. Thus, to obtain

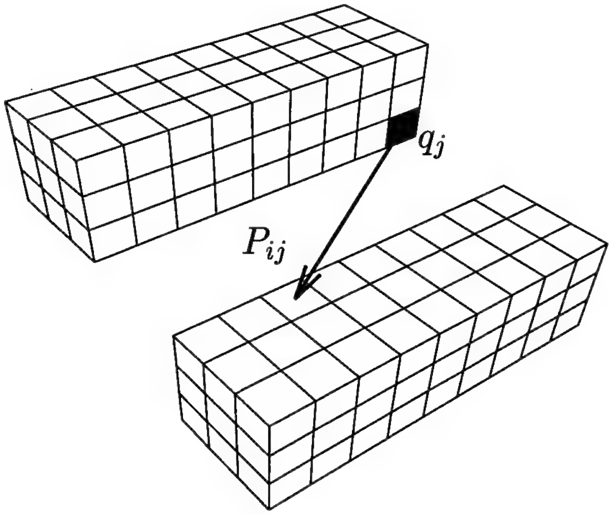


FIGURE 1-1: Piecewise-constant collocation discretization of two conductors. Conductor surfaces are discretized into panels which support a constant charge density.

the capacitance matrix, it is necessary to solve Laplace's equation for a sequence of m Dirichlet boundary conditions. If the conductors are embedded in an infinite homogeneous dielectric, such as free space, a first-kind integral equation may be written[8, 9, 10] for the charge density σ which lies on the conductor surfaces,

$$\psi(x) = \int_{surfaces} \sigma(x') \frac{1}{4\pi\epsilon \|x - x'\|} da' \quad , \tag{1.2}$$

where $\psi(x)$ is the known conductor surface potential, da' is the differential conductor surface area, x, $x' \in \mathbf{R}^3$, ϵ is the dielectric constant, and $\|x\|$ is the Euclidean length of x.

A standard approach[1] to numerically solving (1.2) for the charge density σ is to use a piecewise constant collocation scheme. In such an approach the conductor surfaces are approximated by a set of n polygons, or "panels", and it is assumed that on each panel i, a charge, q_i, is uniformly distributed, as in Figure 1-1. That is, the charge density sigma is approximated by an expansion in basis functions θ_i,

$$\sigma(x) \approx \sum_{i=1}^{n} q_i \theta_i(x), \tag{1.3}$$

where the θ_i are given by

$$\begin{aligned} \theta_i(x) &= 1 \qquad x \in \text{panel } i \\ \theta_i(x) &= 0 \qquad \text{otherwise.} \end{aligned} \tag{1.4}$$

Then for each panel, an equation is written which relates the known potential at the center of that i-th panel, denoted $\overline{f}_i$ and given at the lth potential solution by $\overline{f}_i = V_{kl}$ if panel i is on

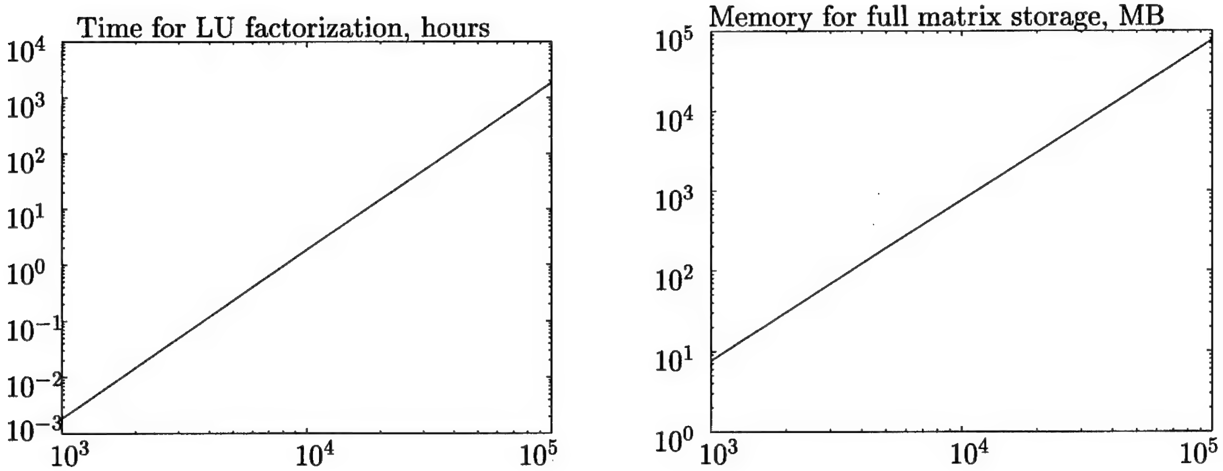


FIGURE 1-2: Computational resources needed by a 100MFLOPS workstation to perform dense LU factorization.

conductor k, to the sum of the contributions to that potential from the n charge distributions on all n panels. The result is the dense linear system,

$$Pq = \overline{f} \tag{1.5}$$

where $P \in \mathbf{R}^{n \times n}$, $q \in \mathbf{R}^n$ is the vector of panel charges, $\overline{f} \in \mathbf{R}^n$ is the vector of known panel potentials, and

$$P_{ij} = \frac{1}{a_j} \int_{panel_j} \frac{1}{4\pi\epsilon \|x_i - x'\|} \, da', \tag{1.6}$$

where the collocation point x_i is the center of the i-th panel and a_j is the area of the j-th panel. For each column l of V, the dense linear system (1.5) is solved to compute the panel charges. The lth column of Q is then obtained by summing the panel charges on each conductor.

The direct approach of solving (1.5) via Gaussian elimination, which requires $O(n^3)$ operations and $O(n^2)$ storage, becomes computationally intractable if the number of panels exceeds several hundred. Figure 1-2 shows that the resources required to solve even a moderately large dense matrix problem by direct methods exceed those available on a modern (circa 1997) scientific workstation. Continuing advances in computer hardware make tractable larger dense matrix problems than were accessible just a few years ago, but the $O(n^3)$ and $O(n^2)$ growth in computational resource requirements ensure that near-$O(n)$ methods will always have substantial performance advantages.

To avoid the formidable computational complexity of Gaussian elimination, an iterative technique such as GMRES [11] may be used to solve (1.5). Normally each iteration of GMRES will cost n^2 operations because the matrix in (1.5) is dense, and therefore evaluating candidate solution vectors involves a dense matrix-vector multiply. The key to accelerating the iterative solution process is the insight that the Pq product can be interpreted as a potential computation.

Numerous algorithms have been proposed for the rapid evaluation of the potential of a set of charges at all the other charge points. Historically the first algorithms developed for this "n-body problem" were the "particle-mesh" methods (see [12] for extensive references) originally concieved for use in simulations in plasma physics. The fast multipole method (FMM)[5], which has enjoyed much recent popularity, and multigrid methods [13], which are probably the most versatile, have also been proposed. Most of these algorithms work by separating the potential into a short-range and a long-range part and then seeking a computationally efficient means to approximate the long-range part of the potential. The various algorithms differ in the way the long-range potential is approximated and in the way local interactions are treated.

The multipole-accelerated iterative approaches[7, 14, 5, 15], now commonly used in a variety of engineering applications [16], require orders of magnitude less storage and computation than algorithms based on direct matrix representations, but are still computationally expensive and memory-exhausting. More importantly, such codes have heretofore been developed only for $1/r$-type kernels. In this work we are concerned with developing algorithms for kernels other than the $1/r$ type generated by free-space Laplace problems. Of primary interest is developing sparse representation procedures for matrices which are derived from potential problems with relatively general kernels, at least including $1/r$ and $\frac{e^{ikr}}{r}$ for a wide range of kr [17, 18, 19, 20, 6, 21].

In this thesis, we describe a precorrected-FFT approach which can replace the fast multipole algorithm for accelerating the potential calculation needed to perform the matrix-vector product in a discretized integral-equation code. The central idea of the algorithm is to represent the long-range part of the potential by point charges lying on a uniform grid, rather than by series expansions as in fast multipole algorithms[5]. This grid representation allows the Fast Fourier Transform (FFT)[22, 23, 24] to be used to efficiently perform potential computations. Because only the long-range part of the potential is represented by the grid, the grid is not coupled to the underlying discretization of the structure. Decoupling the long and short range parts of the potentials allows the algorithm to efficiently solve problems which may be discretized in a very irregular fashion.

We have attempted to develop an algorithm which, like particle-mesh methods, exploits the availability of efficient discrete Fourier transform implementations while at the same time preserves the higher accuracy of the multipole-based schemes, but is also (like multigrid schemes) easily adapted to a broad class of kernels. In addition, the algorithm is, in the way local interactions are treated, particularly adapted to boundary-integral solvers. The hierarchical fast-multipole algorithm [5] can perform the dense matrix-vector product associated with discretized $1/r$ potential integral equations in order-n ($O(n)$) time and memory. The precorrected-FFT method, described below, is at best an $O(n \log n)$ algorithm. It is possible to construct geometries for which the performance of the precorrected-FFT algorithm is inferior to the fast multipole methods, but we demonstrate that for many realistic structures precorrected-FFT methods are faster and use less memory.

This performance improvement and flexibility comes with a cost. First, the precorrected-FFT method can only treat integral equations with a fairly restricted class of kernels – but this is true for most fast integral equation solvers. The kernel must have piecewise-smooth convolutional form. What precisely is meant by piecewise-smooth convolutional form will be made more clear in Chapter 6, and for now it is sufficient to know that both the Laplace and Helmholtz kernel and their close relatives are of this form. Second, problems which are very inhomogeneous, that is, which have regions of densely concentrated field sources interspersed with large regions of empty space, or, in other words, vastly varying densitities of source points, are not treated well by the algorithm. This is perhaps the most glaring shortcoming of the algorithm. Third, the way in which the algorithm accounts for local interactions can be computationally expensive when high accuracy is desired. It is possible to remedy this situation at some cost in accuracy.

On the other hand, the precorrected-FFT method does not suffer from some salient disadvantages of alternative schemes, disadvantages which do impact their practical applicability. Wavelet-type[25] approaches are not applicable for oscillatory kernels. Fast-multipole type algorithms for quasi-static problems do not extend to the Helmholtz case. To the author's knowledge, available fast multipole algorithms for Helmholtz kernels are not numerically stable (or have $O(n^2)$ complexity) for problem domains which are only a few wavelengths in size. The algorithms proposed in [17, 18] require that the size of direct interaction regions be of order of a wavelength, otherwise the operators that diagonalize the multipole translation and conversion operators are not numerically stable. If the problem domain is of the size of a few wavelengths, the number of unknowns contained in the direct interaction regions will be of $O(n)$ making the whole algorithm $O(n^2)$ for a potential computation. Hybrid schemes, similar to the approach proposed in [6] in the context of multigrid methods, are of course possible. Many important problems of engineering interest, such as problems in electromagnetic compatibility analysis and package-level interconnect analysis, fall into this category.

When electromagnetic analysis moves into the non-quasistatic region, an additional dimension is introduced into the analysis, that of frequency-dependence of the matrix elements. Often it is not an analysis at a fixed operating frequency that is required, but the extraction of some frequency-dependent response function.

The simplest method of obtaining the frequency response is to sample the complex Laplace-domain response at discrete points along the imaginary axis. The frequency range of interest may span several decades, however, and the magnitude of the response may also vary over a similar range. Poles close to the imaginary axis can result in very sharp spectral features. Thus, the frequency response may be difficult to resolve by discrete sampling. An alternative approach is to exploit the analytic behavior of the electromagnetic model to construct a rational approximation to the frequency response.

Rational approximation is a useful method to capture the frequency response of a network, but if the approximation is carefully constructed, it can also be used to derive a reduced-order model for use in time-domain circuit simulation. Suppose a network containing both nonlinear elements and large multiport interconnect networks is to be analyzed. An efficient approach is to first reduce the linear interconnect subcircuits to equivalent but much smaller macromodels which reflect the salient behaviour of the original subcircuits. The reduced order models are inserted as a "black box" into the remaining circuit which is then given to simulation.

Methods based on rational approximation, such as asymptotic waveform evaluation (AWE)[26], have been popular for constructing low-order models of circuit interconnect. Algorithms which are based on moments of the frequency response have certain numerical stability problems, but recently the connection between Padé approximation and the Lanczos algorithm[27] has been exploited to obtain orthogonalized Krylov-subspace algorithms which can stably form high-order rational approximants to the frequency response of lumped linear systems[28, 29]. However, one advantage of moment based approaches is that they can construct approximations to systems with *irrational* system response (e.g., a system with delay elements such as transmission lines or the retardation factors of a full-wave electromagnetic model) almost as easily as for a rational response[30, 31], whereas Krylov-subspace model-reduction algorithms for systems with irrational response have yet to be demonstrated.

We begin our exposition in Chapter 2 by discussing some fundamental problems in approximation of potential functions. This discussion leads to the development of the precorrected-FFT algorithm in in Chapter 3. Analysis of the computational complexity of the algorithm, for simple examples, takes place in Chapter 4. In Chapter 5 we discuss application of the algorithm to free-space Laplace problems. Extensions to Laplace problems in stratified geometries and to the scalar Helmholtz equation follow in Chapters 6 and Chapter 7 respectively. Application to full-wave electromagnetic analysis takes place in Chapter 8. Chapter 9 deals with efficient extraction of frequency response functions in the context of full-wave analysis. Finally, in Chapter 10, we discuss some possible extensions of the algorithms presented.

2

Approximation of potential functions

In this chapter the problem of approximating the potential generated by a fixed charge distribution is considered. We derive error bounds for a collocation-grid-projection scheme tuned for use in multilevel methods for solving boundary-element discretizations of potential integral equations. This projection scheme will be used in the algorithm developed in Chapter 3.

2.1 The Approximation Problem

In order to perform an n-charge potential computation in less than quadratic complexity, it is clear that some of the potential interactions must be approximated. The form of approximation chosen will determine many of the features of the ultimate potential computation algorithm.

Consider first the approximation of the potential ϕ of a single point charge located at the origin. The most obvious choice for a function approximation method is polynomial interpolation[32]. Figure 2-1 shows second-order polynomial interpolation applied to the $1/r$ Laplace kernel at two observation points. For sufficiently large r, the approximate (interpolated) potential $\tilde{\phi}$ is a good approximation to the true potential ϕ of a point charge at the origin. The leading order error term is

$$|\phi - \tilde{\phi}| \simeq \frac{1}{r}\left(\frac{h}{r}\right)^3 \tag{2.1}$$

if h is the uniform spacing of the interpolation nodes. This error term becomes large near the singularity at the origin, as can be seen from the inaccurate representation shown in the left side of Figure 2-1.

In more general terms, interpolation is the process of approximating the function $\phi(x)$ by a

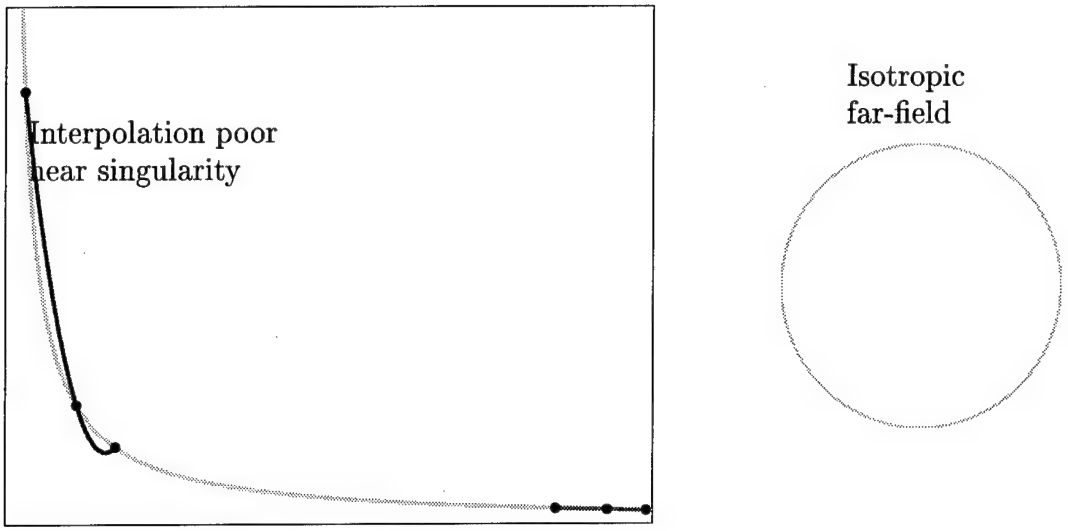


FIGURE 2-1: Left: Interpolation of the $1/r$ Green function. Interpolation is accurate away from the origin but inaccurate near the singularity. Right: Polar plot of the magnitude of the potential of a charge distribution far away from the sources.

weighted sum of tabulated values,

$$\tilde{\phi}(x) = \sum_i w_i \phi(\hat{x}_i) \tag{2.2}$$

where $\hat{x}_i$ are the interpolation nodes, which we will refer to as the *grid points*, or just the grid and the dependence of the weights w_i on x and $\hat{x}_i$ is implied. This process is essentially the same as the problem of approximating the potential of the point charge by the potential of another set of point charges which lie at the gridpoints, as can be seen from the following result [6]:

Theorem 2.1. Given $\tilde{V} \in \mathbf{R}^{m\times 1}$ *is an operator which projects charge onto a grid of* m *points, then* $\tilde{V}^T$ *may be interpreted as an operator which interpolates potential at grid points onto charge coordinates; conversely, given* $\tilde{V}^T \in \mathbf{R}^{1\times m}$ *is an operator which interpolates potential at* m *grid points onto charge coordinates,* $\tilde{V}$ *may be interpreted as an operator which projects charge onto the grid coordinates. In either case,* $\tilde{V}$ *and* $\tilde{V}^T$ *have comparable accuracy.*

Proof. Let $G(x, y)$ be the Green function for a source at y, evaluated at x. Suppose that a unit charge at the point x_0 is represented by the vector of grid charges $\hat{q}$. The approximate potential $\Psi'_x(y)$ at a point y_0 is given by

$$\Psi'_x(y_0) = \sum_i G(y_0, \hat{x}_i)\hat{q} \equiv g^T \hat{q} \tag{2.3}$$

where $\hat{x}_i$ is the position of the ith grid charge and $g \in \mathbf{R}^m$, $g_i = G(y_0, \hat{x}_i)$. Conversely, suppose there is a unit charge at y_0, and the potential $\Psi_y(x_0)$ at x_0 is to be computed by interpolating potentials produced by this unit charge at the grid points $\hat{x}_i$. Then, if $\tilde{V}^T$ is the interpolation operator,

$$\Psi'_y(x_0) = \sum_i \tilde{V}^T(x_0, \hat{x}_i) G(\hat{x}_i, y_0). \tag{2.4}$$

For a symmetric Green function, $\Psi_y(x_0) = G(x_0, y_0) = G(y_0, x_0) = \Psi_x(y_0)$, and

$$\sum_i \tilde{V}^T(x_0, \hat{x}_i) G(\hat{x}_i, y_0) = \tilde{V}^T g \tag{2.5}$$

so that

$$\Psi'_y(x_0) - \Psi_y(x_0) = \tilde{V}^T g - \Psi_y(x_0) = g^T \tilde{V} - \Psi_x(y_0). \tag{2.6}$$

Now suppose $\hat{q}$ is assigned the value $\tilde{V}$ in order to represent the unit point charge at x_0. Then

$$\Psi'_x(y_0) - \Psi_x(y_0) = (g^T q_g) - \Psi_x(y_0) = g^T \tilde{V} - \Psi_x(y_0) = \Psi'_y(x_0) - \Psi_y(x_0). \tag{2.7}$$

□

For a perspective on the relative efficiency of polynomial interpolation, consider expanding the potential into a multipole series[33],

$$\phi(r, \theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{c_{lm}}{r^{l+1}} Y_{lm}(\theta, \phi) \tag{2.8}$$

where $c_{l,m}$ are the multipole expansion coefficients. Suppose for the moment that ϕ is the potential of a point charge located at (a, θ', ϕ'). From the addition theorem[33],

$$\frac{1}{||x - x'||} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{r_<}{r_>^{l+1}} Y^*_{lm}(\theta', \phi') Y_{lm}(\theta, \phi) \tag{2.9}$$

where $x = (r, \theta, \phi), x' = (r', \theta', \phi')$, $r_<$ denotes the smaller of r , r', and $r_>$ the greater, it follows that the $c_{l,m}$ are

$$c_{l,m} = a^l Y^*_{lm}(\theta', \phi'). \tag{2.10}$$

If $\phi(N)$ is the truncation of the series to N terms, i.e.

$$\phi(N) = \sum_{l=0}^{N} \sum_{m=-l}^{l} \frac{c_{lm}}{r^{l+1}} Y_{lm}(\theta, \phi) \tag{2.11}$$

then from bounding the magnitude of the tail of the series, it follows that

$$|\phi - \phi(N)| = O\left[\frac{1}{r}\left(\frac{a}{r}\right)^{N+1}\right]. \tag{2.12}$$

Using order-p polynomial approximation to choose interpolation weights gives errors of order $(a/r)^{p+1}$ and requires $(p+1)^3$ coefficients. Clearly, this is not the most efficient approximation scheme available, as approximations using multipole expansions achieve order $(a/r)^{p+1}$ accuracy with only $(p+1)^2$ coefficients. The convenience of polynomial interpolation, however, is that only the tabulated function values on equally spaced nodes are required for the interpolation/anterpolation operation, as opposed to, e.g., multipole coefficients, which can be awkward to compute for complicated source geometries. It seems reasonable to ask if, for harmonic functions, approximation schemes combining the advantages of both approaches exist. That is, we wish to obtain a scheme which uses tabulated potentials at equally spaced nodes but has better than polynomial accuracy for functions satisfying the Laplace equation.

2.2 The Grid-Potential-Collocation Approach

Consider approximating the potential of a charge distribution $\rho(x)$ by a set of N_G point charges, $Q_j, j = 1 \ldots N_G$ which are positioned at points x_j. Suppose also that both the point charges and the charge distribution lie entirely inside a sphere of radius a centered at the origin. We will require that the potential of the point charges and the potential of the true charge density match at a set of $T \leq N_G$ collocation points $x_{c,\tau}, \tau = 1 \ldots T$ on a closed surface which encompasses the sphere of radius a. That is, for each τ,

$$\sum_j Q_j G(x_j, x_{c,\tau}) = \int \rho(x') G(x', x_{c,\tau}) dx' \tag{2.13}$$

where $G(x, x')$ is the relevant Green's function. It will be convenient if the surface is chosen to be a sphere of radius $r_c > a$, and the collocation points are chosen to be the abscissas of a quadrature rule on the sphere. Integration rules of arbitrary order on a sphere can be constructed by product techniques, but more efficient non-product rules exist [34] which will generally be sufficient for our purposes.

To motivate a scheme for representing panel charges with weighted point charges lying on a grid, consider a charge distribution $\rho(x)$ contained entirely within some small volume B. The potential outside B due to ρ can be determined from the knowledge of the potential on a surface S surrounding B[35]. For example, suppose ρ is contained within a sphere S of radius a. For all (r, θ, ϕ) with $r > a$, the potential ϕ can be written as a multipole expansion series

$$\phi(r, \theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{c_{lm}}{r^{l+1}} Y_{lm}(\theta, \phi) \tag{2.14}$$

where the Y_{lm} are spherical harmonics and the c_{lm} coefficients of expansion. Since the spherical harmonics are orthogonal on a sphere, if the potential is known on any sphere S of radius $r_c > a$, the multipole moments c_{lm} can be computed as

$$c_{lm} = r_c^{l+1} \int_S d\Omega\, Y_{lm}^* \phi(r_c, \theta, \phi). \tag{2.15}$$

The above observation suggests a scheme for computing the grid charges. We presume that the charge whose potential is to be approximated, and the approximating grid charges, lie inside a parallelipiped denoted a "cell." Suppose a $p \times p \times p$ array of grid charges is used to represent

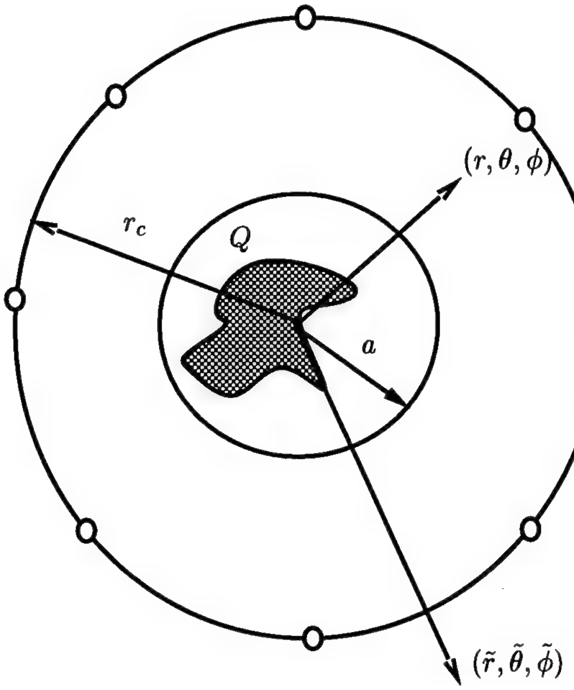


FIGURE 2-2: Potentials $\psi(r < r_c, \theta, \phi)$ or $\psi(\tilde{r} > r_c, \tilde{\theta}, \tilde{\phi})$ for $r, \tilde{r} > a$ may be obtained from the potential at $r = r_c$.

the charge in the cell. First, N_c test points are selected on the surface of a sphere of radius r_c whose center is coincident with the center of the given cell. Then, potentials due to the p^3 grid charges are forced to match the potential due to the cell's actual charge distribution (say m panel charges q) at the test points, i.e.

$$P^{gt}\hat{q} = P^{qt}q \tag{2.16}$$

where $P^{gt} \in \mathbf{R}^{N_c \times p^3}$ is the mapping between grid charges and test point potentials, given by

$$P^{gt}_{i,j} = \frac{1}{\|x^t_i - \hat{x}_j\|}. \tag{2.17}$$

Here x^t_i and $\hat{x}_j$ are the positions of the i-th test point and the j-th grid point, respectively. $P^{qt} \in \mathbf{R}^{N_c \times m}$ is the mapping between panel charges and test point potentials and is given by

$$P^{qt}_{i,j} = \int_{panel\ j} \frac{1}{\|x^t_i - x'\|} da'. \tag{2.18}$$

Since the collocation equation (2.16) is linear in the panel and grid charge distributions, the contribution of the jth panel in the cell to $\hat{q}$ can be represented by a column vector W. W is given by

$$W = \left[P^{gt}\right]^{\dagger} P^{qt,j} \tag{2.19}$$

where $P^{qt,j}$ denotes the jth column of P^{qt} and $[P^{gt}]^{\dagger}$ indicates the generalized Moore-Penrose (or pseudo-) inverse [36] of $[P^{gt}]^{\dagger}$. The computation of $[P^{gt}]^{\dagger}$ is done using the singular value

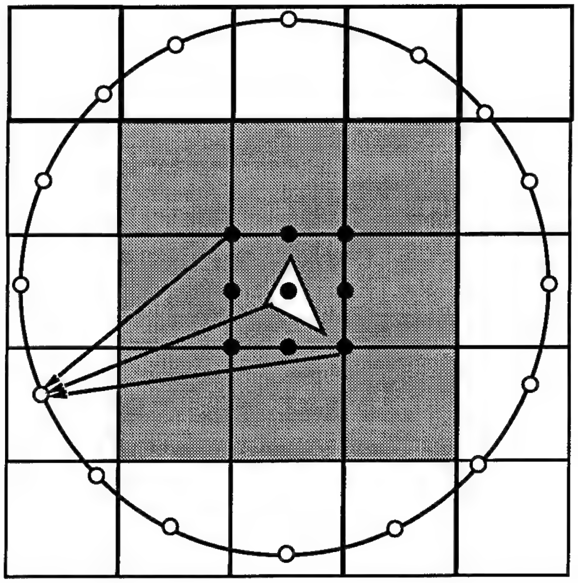


FIGURE 2-3: 2-D Pictorial representation of the grid projection scheme. The black points (at $\hat{x}$) represent the grid charges ($\hat{q}$) being used to represent the triangular panel's charge density σ. The white points are the the points x^t where the potential due to the black point charges and the potential due to the triangular panel's charge density are forced to match. The grid charges approximate the panel potential outside the gray region.

decomposition. The error analysis below will depend on the boundedness of W. For arbitrary order of approximation and quadrature rules, $[P^{gt}]^\dagger$ is unbounded,[1] but for the relatively low order rules of interest in this thesis, it is a simple matter to check that

$$\kappa \equiv \left\| \left[P^{gt}\right]^\dagger P^{qt,j} \right\| \tag{2.20}$$

remains small. The grid projection scheme is summarized in Figure 2-3.

The accuracy of the above projection scheme hinges on the proper selection of the test points x^t. From Equation 2.15, we expect high accuracy if the test points are chosen to be abscissas of a high-order quadrature rule[34]. It will be shown that the error in potential due to the grid-charge approximation of a charge distribution contained within a sphere of radius a, at a distance r from the center of the distribution, is of order $(a/r)^{(M+1)/2}$ if the test points are chosen to be the nodes of a quadrature rule accurate to order M[38].

2.3 Error analysis

First we establish error bounds for the approximation of a panel charge potential by grid charges.

---

[1] This follows from results in [37] and Equation 2.29 below.

Theorem 2.2. Suppose a grid-charge representation of a charge distribution $\rho(\vec{x})$, of total charge $Q = \int |\rho(\vec{x}')| d\vec{x}'$, lying inside a sphere $S(a)$ of radius a centered at the origin, has been constructed. Assume the grid charges Q_j are given at points $\vec{x}_j$, $j = 1 \ldots N_G$, and define $Q_G = \sum_{j=1}^{N} |Q_j|$. It is possible to choose a radius r_c of the collocation sphere such that the error ϕ_e in the grid-charge approximation of the potential in the $k = 0$ case satisfies

$$|\phi_e| \le \frac{Q + Q_G}{r} \left(\frac{a}{r}\right)^{\frac{M+1}{2}} \frac{M^2 + M + 1}{1 - (a/r_c)} + \frac{Q + Q_G}{r} \left(\frac{a}{r}\right)^{M+1} \frac{1}{1 - (a/r)} \qquad (2.21)$$

where M is the order of the quadrature rule and r is the distance from the evaluation point to the origin.

Proof. The multipole expansion of potential ϕ of the charge distribution is [33]

$$\phi(r, \theta, \phi) = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{1}{2l+1} \frac{1}{r^{l+1}} [\int_{S(a)} r'^{l} \rho(\vec{x}') Y_{lm}^{*}(\theta', \phi') d\vec{x}'] Y_{lm}(\theta, \phi). \qquad (2.22)$$

Similarly, the multipole expansion of the grid-charge potential $\phi_g(r, \theta, \phi)$ is

$$\phi_g(r, \theta, \phi) = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{1}{2l+1} \frac{1}{r^{l+1}} [\sum_{j=1}^{N_G} Q_j r_j^l Y_{lm}^{*}(\theta_j, \phi_j)] Y_{lm}(\theta, \phi). \qquad (2.23)$$

Let $(r_c, \theta_\tau, \phi_\tau)$ denote the τth collocation point, $\tau = 1 \ldots T$, on the surface of the sphere of radius r_c. Assume that the (θ_τ, ϕ_τ) are the abscissas of a quadrature rule on a sphere such that the rule exactly integrates spherical polynomials of degree at least M. Let $w_\tau, \tau = 1 \ldots T$ denote the quadrature rule weights corresponding to a sphere of radius unity.

The error in the potential, ϕ_e, may be expanded in multipole series,

$$\phi_e(r, \theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{q_{l,m}}{r^{l+1}} Y_{lm}(\theta, \phi) \qquad (2.24)$$

where the $q_{l,m}$ are the multipole expansion coefficients given by

$$q_{l,m} = \left[\frac{4\pi}{2l+1} \int_{S(a)} r'^{l} \rho(\vec{x}') Y_{lm}^{*}(\theta', \phi') d\vec{x}' - \sum_{j=1}^{N_G} Q_j r_j^l Y_{lm}^{*}(\theta_j, \phi_j) \right]. \qquad (2.25)$$

The collocation condition can be used to put bound on the magnitude on the low-order $q_{l,m}$. At a collocation point, the error in the potential, $\phi_e(r_c, \theta_\tau, \phi_\tau) = \phi(r_c, \theta_\tau, \phi_\tau) - \phi_g(r_c, \theta_\tau, \phi_\tau)$ is zero, so we may write

$$\sum_{l=0}^{M} \sum_{m=-l}^{l} \frac{1}{r_c^{l+1}} q_{l,m} Y_{lm}(\theta_\tau, \phi_\tau) = \qquad (2.26)$$

$$4\pi \sum_{l=M+1}^{\infty} \sum_{m=-l}^{l} \frac{1}{2l+1} \frac{1}{r_c^{l+1}} \left[\int_{S(a)} r'^{l} \rho(\vec{x}') Y_{lm}^{*}(\theta', \phi') d\vec{x}' - \sum_{j=1}^{N_G} Q_j r_j^l Y_{lm}^{*}(\theta_j, \phi_j) \right] Y_{lm}(\theta_\tau, \phi_\tau)$$

for $\tau = 1 \ldots T$.

From the identity

$$\int d\Omega \, Y^*_{l'm'}(\theta, \phi) Y_{lm}(\theta, \phi) = \delta_{ll'}\delta_{mm'}, \tag{2.27}$$

and the quadrature rule for selecting $w_\tau, \theta_\tau, \phi_\tau$, it then follows that

$$\sum_\tau w_\tau Y^*_{l'm'}(\theta_\tau, \phi_\tau) Y_{lm}(\theta_\tau, \phi_\tau) = \delta_{ll'}\delta mm' \tag{2.28}$$

for $l + l' \leq M$. Therefore, multiplying each side of (2.26) by $w_\tau Y^*_{l'm'}(\theta_\tau, \phi_\tau)$ and summing over τ leads to a simplified form,

$$\frac{1}{r_c^{l'+1}} q_{l',m'} = \tag{2.29}$$

$$4\pi \sum_{\tau=1}^{T} w_\tau \sum_{l=M+1-l'}^{\infty} \sum_{m=-l}^{l} \frac{1}{2l+1} \frac{1}{r_c^{l+1}} \left[\int_{S(a)} r'^l \rho(\vec{x'}) Y^*_{lm}(\theta', \phi') d\vec{x'} - \sum_{j=1}^{N_G} Q_j r_j^l Y^*_{lm}(\theta_j, \phi_j) \right] Y^*_{l'm'}(\theta_\tau, \phi_\tau) Y_{lm}(\theta_\tau, \phi_\tau)$$

The addition theorem for spherical harmonics states

$$\frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y^*_{l,m}(\theta', \phi') Y_{l,m}(\theta, \phi) = P_l(\cos\gamma) \tag{2.30}$$

where γ is the angle between (θ', ϕ') and (θ, ϕ) and $P_l(\cos\gamma)$ a Legendre polynomial.

Using the addition theorem to collapse the spherical harmonics, we have

$$\phi_e(N) \equiv \sum_{l'=0}^{N} \sum_{m'=-l'}^{l'} \frac{1}{r^{l'+1}} q_{l',m'} Y_{l'm'}(\theta, \phi) = \tag{2.31}$$

$$\sum_{\tau=1}^{T} w_\tau \sum_{l'=0}^{M} \sum_{l=M+1-l'}^{\infty} \frac{2l'+1}{4\pi} \frac{1}{r^{l'+1}} \frac{r_c^{l'+1}}{r_c^{l+1}} [\int_{S(a)} r'^l \rho(\vec{x'}) P_l(\alpha') d\vec{x'} - \sum_{j=1}^{N_G} Q_j r_j^l P_l(\beta_j)] P_{l'}(\gamma_\tau)$$

for some angles α, β, γ and $N < M + 1$.

Since $|P_l(\cos\eta)| \leq 1$ for any η,

$$|\phi_e(N)| \leq \frac{Q + Q_G}{r} \sum_{\tau=1}^{T} w_\tau \sum_{l'=0}^{M} \sum_{l=M+1-l'}^{\infty} \frac{2l'+1}{4\pi} \left(\frac{r_c}{r}\right)^{l'} \left(\frac{a}{r_c}\right)^{l} \tag{2.32}$$

and since $\sum_\tau w_\tau = 4\pi$

$$|\phi_e(N)| \leq \frac{Q + Q_G}{r} \sum_{l'=0}^{N} (2l'+1) \left(\frac{r_c}{r}\right)^{l'} \left(\frac{a}{r_c}\right)^{M+1-l'} \frac{1}{1 - (a/r_c)} \tag{2.33}$$

The total error is the sum of $\phi_e(N)$ and the contribution from the neglected multipole terms of orders $> N$. Various bounds can be obtained by manipulating the choices of N and r_c in Eq. 2.33. It is particularly instructive to consider $N = M, r_c = \sqrt{ar}$, in which case

$$\frac{a}{r_c} = \frac{r_c}{r} = \sqrt{\frac{a}{r}} \tag{2.34}$$

so that

$$|\phi_e(N)| \leq \frac{Q + Q_G}{r} \left(\frac{a}{r}\right)^{\frac{M+1}{2}} \frac{M^2 + M + 1}{1 - (a/r_c)} \tag{2.35}$$

Adding in the contribution from the neglected terms in the error series, we have a final bound for the error ϕ_e:

$$|\phi_e| \leq \frac{Q + Q_G}{r} \left(\frac{a}{r}\right)^{\frac{M+1}{2}} \frac{M^2 + M + 1}{1 - (a/r_c)} + \frac{Q + Q_G}{r} \left(\frac{a}{r}\right)^{M+1} \frac{1}{1 - (a/r)} \tag{2.36}$$

□

We now have the main result of this section.

Theorem 2.3. Suppose the potential of a point charge is given by $1/r$. *The grid-based technique for evaluating, outside a sphere of radius* r_m, *the potential of a charge density of total charge magnitude* Q, *located inside a sphere of radius* a, *has error* ϕ_e *bounded by*

$$|\phi_e| \leq (1 + \kappa) \frac{Q}{r_m} (\frac{a}{r_m})^{\frac{M+1}{2}} \frac{(M+1)^2}{1 - \sqrt{(a/r_m)}} \tag{2.37}$$

where M *is the order of a quadrature rule on a sphere and* r_m *is the distance of the nearest potential-evaluation evaluation point to the origin,* $r_m > a$ *and is defined in Eq. 2.20.*

Proof. The theorem follows directly from Theorems 2.1 and 2.2. □

Theorem 2.3 provides an error bound and thus a convergence proof, but the bound given is not especially useful as an error estimator in empirical studies, as it was derived for r_c a function of r and a. To obtain more appropriate results it is necessary to divide the problem into two regions of study: $r_c > r$ and $r_c < r$. We shall see empirically and analytically that the error behavior of the grid-collocation method is qualitatively different in the two regimes. It is a simple matter to obtain the following from Equation 2.33, for fixed r_c:

$$|\phi_e| \leq \frac{Q + Q_G}{r} \left(\frac{a}{r_c}\right)^{\frac{M+1}{2}} \frac{M^2 + M + 1}{1 - (a/r_c)} + \frac{Q + Q_G}{r} \left(\frac{a}{r}\right)^{\frac{M+3}{2}} \frac{1}{1 - (a/r)} \qquad r_c < r \tag{2.38}$$

$$|\phi_e| \leq \frac{Q + Q_G}{r} \left(\frac{a}{r}\right)^{\frac{M+1}{2}} \frac{M^2 + M + 1}{1 - (a/r_c)} + \frac{Q + Q_G}{r} \left(\frac{a}{r}\right)^{\frac{M+3}{2}} \frac{1}{1 - (a/r)} \qquad r_c > r \tag{2.39}$$

As a function of r, we expect the errors to decrease sharply only until $r \simeq r_c$, afterwards it is the aliasing errors that dominate the approximation.

2.4 Helmholtz Kernels

Consider now approximating the potential of a point source with Green function e^{ikr}/r. Figure 2-4, analogous to 2-1, shows interpolation of such a kernel. On the left of Figure 2-4 we see that the introduction of the new length scale $\lambda = 2\pi/k$ has made the interpolation inaccurate away from the singularity, because the node spacing is insufficiently small compared to λ. On the right is shown the intensity of the potential for one possible configuration of sources, for a large value of ka where a is the dimension of the source region. We see that the potential can take on arbitrarily complicated patterns far from the sources. This is different from the $1/r$ case where, sufficiently far from a source region, the potential is always isotropic (the potential tends to the $1/r$ monopole). To accurately approximate the potential these patterns must be resolved.

It is important to distinguish between the one- and many- dimensional cases when discussing accuracy of various interpolation schemes. In one dimension, polynomial interpolation requires $p+1$ coefficients to achieve order-p accuracy. In three dimensional, polynomial interpolation can be done by a sequence of one-dimensional interpolations, and just as for the $1/r$ kernel, $(p+1)^3$ coefficients are needed for order-p accuracy. But again, using multipole expansions, $(p+1)^2$ coefficients are needed for order-p accuracy. The exponent of two comes from the fact that the multipole expansions efficiently represent the far-field radiation patterns, which live on a two-dimensional surface. Polynomials, in contrast, must sample the three-dimensional fields on a scale of λ. It is hoped that the grid-collocation method can preserve the efficiency of multipole representations by matching the far-field data. In this section, we will show that, just as for the $1/r$ case, the grid-collocation method obtains accurate approximations to the low order multipole coefficients, which is equivalent to low-order interpolation of the far-field patterns. Section 2.5 discusses modifications to the grid-collocation method that directly utilize the far-field data.

In the previous section we obtained error bounds for the grid-collocation technique using a three-step process. The first step was to bound the sum of a tail of a multipole expansion series. The second was to use the collocation condition to obtain an expression for the multipole coefficients of the error in terms of a series tail. Finally, by bounding the sum of this "aliasing" error due to the errors in the low-order multipole coefficients, an error bound for the grid-collocation technique was obtained. A similar approach can be followed when the kernel is e^{ikr}/r.

In the low-frequency limit, $k \to 0$, the asymptotic expressions[39]

$$z^{-n} j_n(z) \to \frac{1}{1 \cdot 3 \cdot 5 \ldots (2n+1)} \tag{2.40}$$

$$z^{n+1} y_n(z) \to -1 \cdot 3 \cdot 5 \ldots (2n-1) \tag{2.41}$$

make the extension trivial. Additionally, the spherical Bessel functions are known to approach

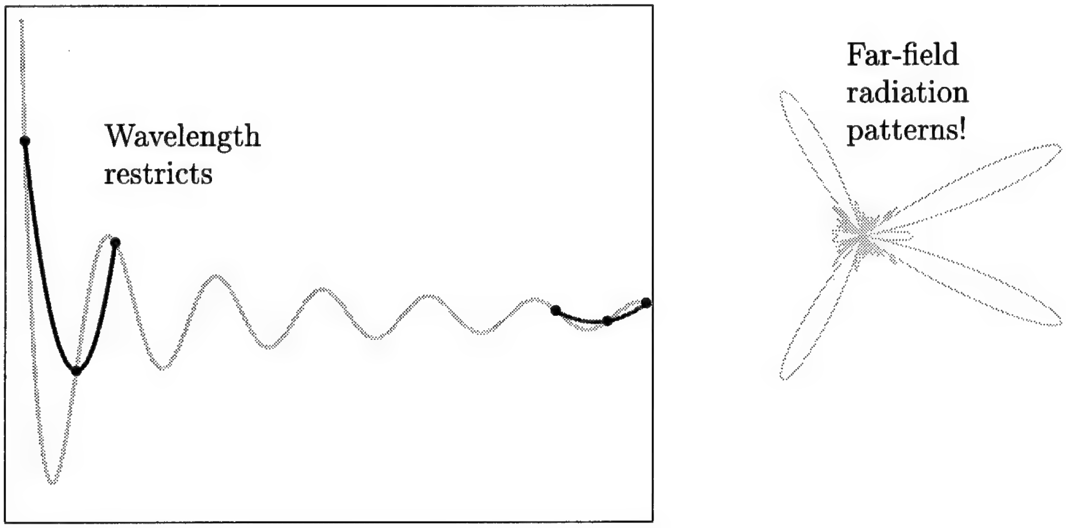


FIGURE 2-4: Left: Interpolation of the real part of the e^{ikr}/r Green function. Interpolation is inaccurate near the singularity as in the $1/r$ case, but may also be inaccurate away from the singularity if the spacing of the interpolating points is large compared to the characteristic wavelength $2\pi/k$. Right: Polar plot of the magnitude of the potential of a hypothetical charge distribution in the far-field limit.

the asymptotic limits[39, 18]

$$\lim_{n\to\infty}\left| j_n(z)\frac{2(2n+1)^{n+1}}{z^n e^{n+1/2}}\right| = 1 \tag{2.42}$$

$$\lim_{n\to\infty}\left| h_n(z)\frac{z^{n+1}e^{n+1/2}}{\sqrt{2}(2n+1)^n}\right| = 1 \tag{2.43}$$

and from these expressions we can draw similar conclusions, as long as the order of the quadrature rule is sufficiently large. For larger kL, with L the size of the computational region, and moderate order M, however, the situation is more difficult.

The following bounds, which can be obtained from 10.1.13 and 10.1.16 in [39], will be needed:

$$|j_n(z)| \leq \frac{\pi}{2}\frac{z^n}{2^n n!} \tag{2.44}$$

$$|h_n(z)| \leq \frac{1}{z}\left(1+\frac{n}{z}\right)^n. \tag{2.45}$$

Consider first the error present in truncating a multipole expansion of a Helmholtz field to N terms. As is the usual case, suppose that outside a sphere of radius a, a function ψ satisfying the Helmholtz equation is represented by a multipole expansion whose moments up to order N vanish

$$\psi(r,\theta,\phi) = \sum_{l=N+1}^{\infty}\sum_{m=-l}^{l} p_{l,m}h_l^{(1)}(kr)Y_{lm}(\theta,\phi) \tag{2.46}$$

where k is the wavenumber and $h_l^{(1)} = j_l(kr) + iy_l(kr)$ is the first-kind spherical Hankel function of order l.

Theorem 2.4. For $N > ka$ and any $r > a$, there exist constants $c_1, c_2 > 0$, independent of a, r and k, such that

$$|\psi(r, \theta, \phi)| \leq c_1 (\frac{a}{r})^{N+1} + c_2 \left(\frac{ka}{N+1}\right)^{N+1} \tag{2.47}$$

Proof. Follows directly from Eqs. 2.44 and 2.45. □

The second term occurs due to the fact that, regardless of how far the potential evaluation point is from the source, the complex far-field radiation pattern must be resolved. We expect that as long as the order of the quadrature rule used in the grid-collocation technique is larger than about $2ka$, it should be possible to accurately determine all the significant multipole coefficients of the potential.

Proposition 2.5. Suppose the potential of a charge is given by e^{ikr}/r. If the collocation points in the grid-charge assignment are chosen to be the abscissas of a quadrature rule which exactly integrates spherical harmonics of order $\leq 2ka$, i.e.,

$$M > 2ka \tag{2.48}$$

for a quadrature rule of order M, then the grid-based technique for evaluating, outside a sphere of radius r_m, the potential of a charge density of total charge magnitude Q, located inside a sphere of radius a, has error ϕ_e bounded by

$$|\phi_e| \leq c_1(1+\kappa)\frac{Q}{r_m}(\frac{a}{r_m})^{\frac{M+1}{2}} \frac{(M+1)^2}{1-\sqrt{a/r_m}} + c_2 \left(\frac{ka}{M+1}\right)^{M+1} \tag{2.49}$$

for some constants c_1, c_2 independent of a, r, k.

Proof. (*Sketch*) Given the conditions of Theorem 2.4, the proof follows similarly to the proof of Theorem 2.3.

The multipole expansion of potential ϕ of the charge distribution is [33]

$$\phi(r, \theta, \phi) = 4\pi ik \sum_{l=0}^{\infty} \sum_{m=-l}^{l} h_l(kr)[\int_{S(a)} j_l(kr')\rho(\vec{x'})Y_{lm}^*(\theta', \phi')d\vec{x'}]Y_{lm}(\theta, \phi). \tag{2.50}$$

and the multipole expansion of the grid-charge potential $\phi_g(r, \theta, \phi)$ is

$$\phi_g(r, \theta, \phi) = 4\pi ik \sum_{l=0}^{\infty} \sum_{m=-l}^{l} h_l(kr)[\sum_{j=1}^{N_G} Q_j j_l(kr_j)Y_{lm}^*(\theta_j, \phi_j)]Y_{lm}(\theta, \phi). \tag{2.51}$$

The condition that the error $\phi_e(r_c,\theta_\tau,\phi_\tau) = \phi(r_c,\theta_\tau,\phi_\tau) - \phi_g(r_c,\theta_\tau,\phi_\tau)$ in the potential at a collocation point $(r_c,\theta_\tau,\phi_\tau)$ is zero is

$$\sum_{l=0}^{M}\sum_{m=-l}^{l} h_l(kr_c)q_{l,m}Y_{lm}(\theta_\tau,\phi_\tau) = \tag{2.52}$$

$$4\pi ik \sum_{l=M+1}^{\infty}\sum_{m=-l}^{l} h_l(kr_c)\left[\int_{S(a)} j_l(kr')\rho(\vec{x'})Y^*_{lm}(\theta',\phi')d\vec{x'} - \sum_{j=1}^{N_G} Q_j j_l(kr_j)Y^*_{lm}(\theta_j,\phi_j)\right] Y_{lm}(\theta_\tau,\phi_\tau)$$

for $\tau = 1\ldots T$, where $q_{l,m}$ is given by

$$q_{l,m} = \left[4\pi ik \int_{S(a)} j_l(kr')\rho(\vec{x'})Y^*_{lm}(\theta',\phi')d\vec{x'} - \sum_{j=1}^{N_G} Q_j j_l(kr_j)Y^*_{lm}(\theta_j,\phi_j)\right]. \tag{2.53}$$

From this the low-order multipole coefficients $q_{l',m'}$ of the error are,

$$h_{l'}(kr_c)q_{l',m'} = \tag{2.54}$$

$$4\pi ik \sum_{\tau=1}^{T} w_\tau \sum_{l=M+1}^{\infty}\sum_{m=-l}^{l} h_l(kr_c)\left[\int_{S(a)} j_l(kr')\rho(\vec{x'})Y^*_{lm}(\theta',\phi')d\vec{x'} - \right.$$

$$\left.\sum_{j=1}^{N_G} Q_j j_l(kr_j)Y^*_{lm}(\theta_j,\phi_j)\right] Y^*_{l'm'}(\theta_\tau,\phi_\tau)Y_{lm}(\theta_\tau,\phi_\tau)$$

and after collapsing the spherical harmonics, the expression for the error in the potential $\phi_e(N)$ due to the low-order multipole coefficients is

$$\phi_e(N) = ik\sum_{l'=0}^{N}\sum_{m'=-l'}^{l'} h_{l'}(kr)q_{l',m'}Y_{l'm'}(\theta,\phi) = \tag{2.55}$$

$$\sum_{\tau=1}^{T} w_\tau \sum_{l'=0}^{N}\sum_{l=M+1-l'}^{\infty} \frac{(2l'+1)(2l+1)}{4\pi} h_{l'}(kr)\frac{h_l(kr_c)}{h_{l'}(kr_c)}\left[\int_{S(a)} j_l(kr')\rho(\vec{x'})P_l(\alpha')d\vec{x'} - \sum_{j=1}^{N_G} Q_j j_l(kr_j)P_l(\beta_j)\right]P_{l'}(\gamma_\tau)$$

for some angles α,β,γ.

Thus, the expression analogous to Eq. 2.32 which must be bounded is

$$|\phi_e(N)| \le \left|\frac{Q+Q_G}{r}\sum_{l'=0}^{N}\sum_{l=M+1-l'}^{\infty}(2l'+1)(2l+1)k\frac{h_{l'}(kr)}{h_{l'}(kr_c)}j_l(ka)h_l(kr_c)\right| \tag{2.56}$$

If we again suppose, for example, $N = M, r_c = \sqrt{ar}$, in which case

$$\frac{a}{r_c} = \frac{r_c}{r} = \sqrt{\frac{a}{r}} \tag{2.57}$$

so $r > r_c$, and additionally require $kr_c > M$, then it is straightforward to show

$$\phi_e(N) \le c'_1\left(\frac{a}{r}\right)^{\frac{M+1}{2}} + c'_2\left(\frac{ka}{M+1}\right)^{\frac{M+1}{2}} \tag{2.58}$$

for some c'_1, c'_2. □

2.5 Utilizing the far-field expressions

Potentials generated by a Helmholtz kernel possess structure in the far-field ($r \to \infty$) limit that can be used to represent the potentials in much the same way as the multipole coefficients. Essentially, the multipole coefficients are Fourier coefficients for the expansion of the far-field functions in a basis of spherical harmonics. The far-field potential of a charge at position x' is

$$F(s) = e^{|x'| c(x', s)} \tag{2.59}$$

where $s \in S^2$, S^2 is the unit sphere, and $c(x', s)$ is the cosine of the angle between x and s.

The potential of the charge at the point $(r, \Omega), \Omega \in S^2$, can be determined for $r > |x'|$ from the far-field potentials by [18]

$$\phi = \lim_{n \to \infty} \int_{S^2} \sum_{l=0}^{n} i^l (2l+1) h_l(kr) P_l(c(\Omega, s)) \tag{2.60}$$

This suggests that a good way to calculate the grid-projection operators might be to require that the far-field potential of the grid-charges and the actual charge distribution match at the collocation points $s_\tau \in S^2$. Let the far-field of the error ϕ_e in the potential be given by

$$F(s) = \sum_{l', m'} q_{l', m'} Y_{l', m'}(s) \tag{2.61}$$

From the definition of the far-field expansion (Equation 2.59) and 10.1.47 in [39] it follows that

$$F(s) = \sum_{l'=0}^{\infty} \sum_{m'=-l}^{l} \left[\sum_j q_j j_{l'}(kr_j) Y^*_{l', m'}(\omega_j) - \int \rho(x') j_{l'}(kr') Y^*_{l', m'}(\omega') \right] Y_{l', m'}(s) \tag{2.62}$$

where $\omega_j, \omega \in S^2$ are the angular coordinates of the jth grid-charge a source charge point. Then, from the collocation condition it follows that

$$q_{l', m'} = \sum_{l=M+1-l'}^{\infty} \sum_{m=-l}^{l} \sum_{\tau} w_\tau \sum_j q_j j_l(ka_j) Y^*_{lm}(w_j) \qquad l' < M+1 \tag{2.63}$$

From this and Eq. 2.45, it follows in a manner similar to the preceding proofs that

$$|\phi_e| c_2 \left(\frac{a}{r}\right)^{\frac{M+1}{2}} + \le c_1 \left(\frac{ka}{M+1}\right)^{\frac{M+1}{2}}. \tag{2.64}$$

2.6 Applications and competing approaches

While the grid operators described here were developed with the precorrected-FFT technique in mind, they can be incorporated into any multi-level scheme [21, 6]. The representation described here has two advantages which allow it to be efficient. First, because of the regular spacing of the grid charges, fast ($O(l^2) \log l$, where l is the order of the quadrature rule) translation and potential evaluation operators exist. It appears that in the approach in [21], only the $O(l^4)$ direct operators are available. Secondly, the sharing of grid charges between computational cells allows for a reduction in the total number of coefficients needed to represent the potential in each cell of the computational domain. That is, if there are N cells in the domain, and p^3 grid charges are used to represent the potential in each cell, then, for large N where we may neglect edge effects, the total number of grid charges is only $N(p-1)^3$, a significant reduction for small p. For most engineering problems, we expect $p \leq 5$, so the sharing effect will still be significant. An additional advantage of the grid-based approach is that the potential *throughout* the domain can be obtained at little additional cost once the panel charges have been determined [40].

Many other approaches to computing the grid charge are available. For an alternative approach, based on matching multipole expansion coefficients directly, see [37]. [6] discusses deriving coefficients from polynomial interpolation.

2.7 Empirical Grid Error Analysis

First we consider the worst-case errors in the grid-charge representation, and compare to representation via multipole expansions. For an order-L multipole expansion ϕ_m, the bound on the error $\phi - \phi_m$ in the approximation is

$$|\phi - \phi_m| \leq \frac{Q}{r}\left(\frac{a}{r}\right)^{L+1}\frac{1}{1-(a/r)} \tag{2.65}$$

while for a grid-charge representation ϕ_g , collocating at nodes of an order M quadrature rule, the error must satisfy

$$|\phi - \phi_g| \leq \frac{Q}{r}\left(\frac{a}{r}\right)^{(M+1)/2}\frac{(M+1)^2}{1-\sqrt{a/r}}(1+\kappa) \tag{2.66}$$

For an order-M quadrature rule, it seems reasonable to define an effective order of the grid-charge error $N_g = (M-1)/2$. Figure 2-5 shows the error vs. effective order for multipole expansions and the grid scheme, assuming cell interactions up to first-near-neighbors will be done exactly, for some difficult sample charge distributions. The worst-case error for multipole expansions appears to often occur for a charge located at the corner of the unit cube, and an evaluation point projected onto the nearest face of the near-neighbor cube boundary. This source-charge/test-point configuration was used for the calculations of multipole expansion errors in Figures 2-5 and 2-7. The sample charge for the evaluation of the grid-charge was placed sometimes at the corner, and sometimes on the cube edge halfway between two grid charge points, whichever generated the largest error, with the evaluation point again at the nearest point of the near-neighbor cube boundary. For the same order, the multipole and grid-charge representations have comparable errors, indicating that the bound of Eq. 2.66 is relatively loose. Thus to obtain an idea of the relative efficiencies of the two approaches, it is reasonable

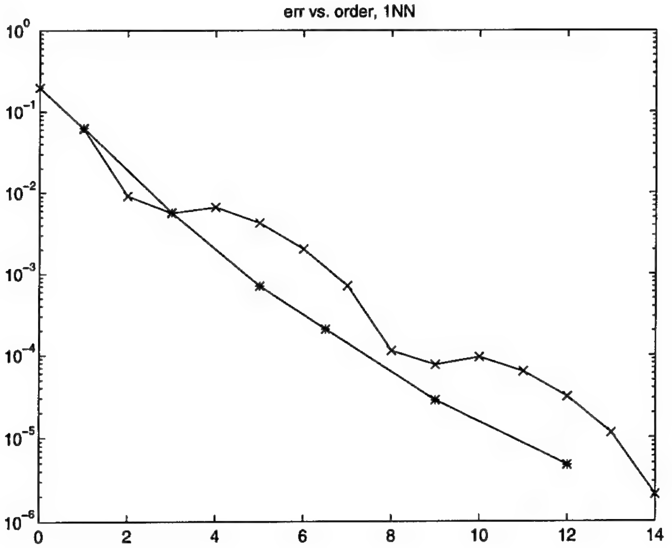


FIGURE 2-5: Error vs. effective order for multipole expansions (x) and grid-projection scheme (*), at first-near-neighbors cells. For an order-M quadrature rule, the effective order of the grid-projection scheme is $(M-1)/2$.

to compare the number of coefficients per unit cell required to obtain a given order of accuracy. For an order-L multipole expansion, $(L+1)^2$ coefficients are needed. For a $p \times p \times p$ grid-charge representation, there are $(p-1)^3$ charges per unit cell. The order of the approximation depends on the quadrature rule used, but in general an efficient rule with order $M \simeq 2\lfloor\sqrt{p^3/2}\rfloor - 1$ is available. From Figure 2-6, which shows the number of coefficients required for both schemes, it is clear that in terms of the effective order, the grid-charge representation is more efficient at low orders. This is due both to the greater amount of charge-sharing at low orders, and to the availability of efficient non-product integration rules. Figure 2-7 shows that in terms of the effective number coefficients per cell, the grid-based scheme may have superior worst-case error performance up to at least order 6 or so.

Next, we consider the behavior of errors away from the charge distribution. Consider placing a single point charge in the basic computational cell and using the various projection schemes to approximate its potential via sets of point charges at the regular grid points. If the cell is a cube with side length d, empirical analysis suggests that the grid-collocation projection scheme has largest error when the point charge is located on the face of the cube with coordinates $(d/2)(1-1/(p-1), 1, 1-1/(p-1))$, with the evaluation point taken at a point along a line parallel to the y-axis and intersecting the test charge, i.e. $(d/2)(1-1/(p-1), 1+\Delta y, 1-1/(p-1))$, for some increment Δy. Figures 2-8 to 2-11 show errors at various values of r_c and kd, with k the wavenumber. Errors for the use of polynomial interpolation and, for $k > 0$, the far-field collocation approach are also shown.

Consider first the (b) plots, where r_c is small. For all orders of approximation the error decays slowly away from the charge distribution. Since in this case $r_c \simeq r_{min}$, from the error

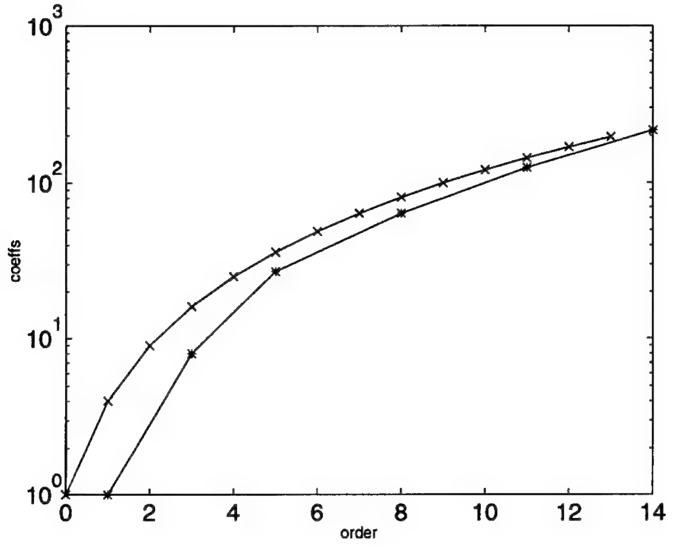


FIGURE 2-6: Effective number of coefficients per cell vs. effective order for multipole expansions (x) and grid-projection scheme (*).

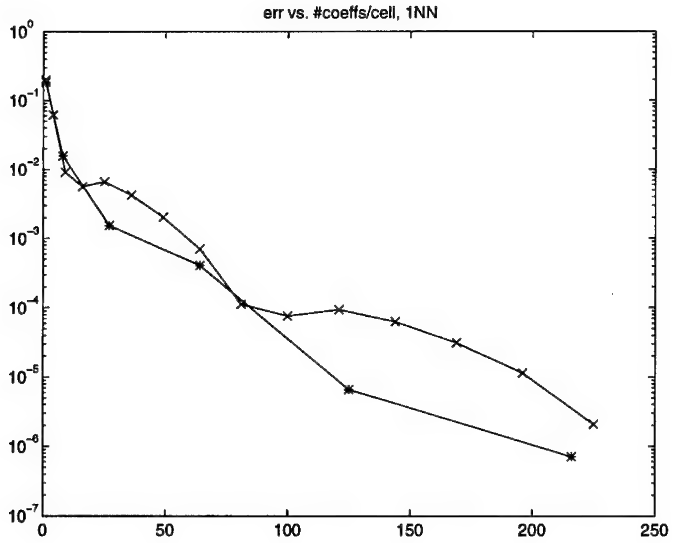


FIGURE 2-7: Error vs. effective number of coefficients per cell for multipole expansions (x) and grid-projection scheme (*), at first-near-neighbor cells.

p	Rule Order M	Source	max kd
$p = 2$	3	[41]	1.5
$p = 3$	7	[41]	3
$p = 4$	11	[41]	5
$p = 5$	17	[42]	8
$p = 6$	23	[43]	10

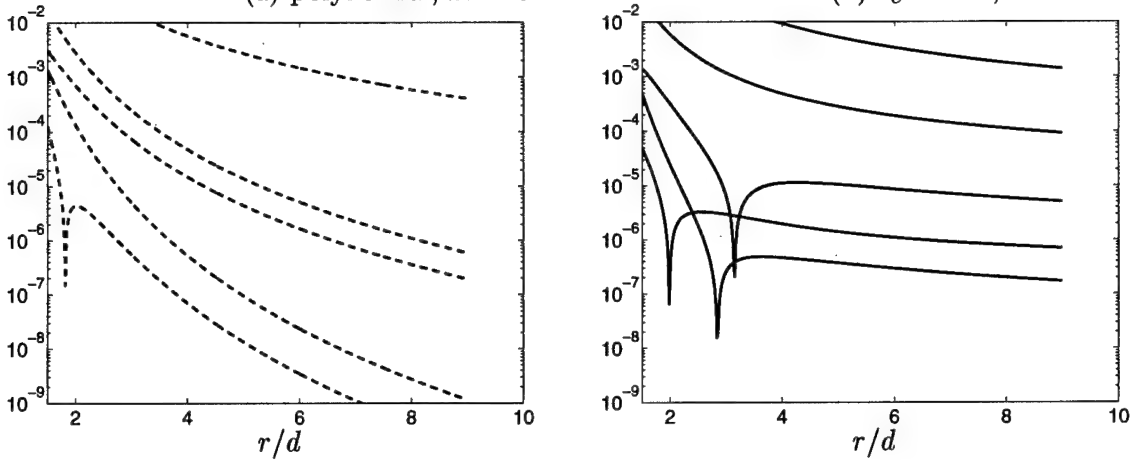
Table 2-1: Quadrature rules used for the grid-projection scheme experiments.

analysis of Eqs. 2.38 and 2.39, we expect the error to behave essentially as a monopole, dying slowly away from the origin, regardless of the order of the quadrature rule. We only expect the order of quadrature rule to change the constant factor in front of the error term. Notice that in general, when r_c is increased in the (c)-(e) plots, the worst-case errors do not change drastically, again as is predicted by our previous analysis. The variation of error with distance, however, changes considerably. As the collocation sphere radius is increased, the magnitude of the low order multipole coefficients of the error decreases, and the errors decay rapidly with distance. Note that the sharp error decay associated with high order multipole approximation ends at about the collocation sphere radius.

As k is increased in Figures 2-9 to 2-11, the various approximation methods all become less accurate, and the low-order schemes become totally inaccurate. Table 2-1 shows the expected value of kd at which the collocation scheme should fail for each order, based on the $M \geq 2kd$ criterion. Our results are in good agreement with these estimates. In particular the $p = 6$ collocation scheme retains good accuracy even at the highest frequency considered. Since the order of the collocation rule (23 in this case) is much greater than $2kd$, we expect the errors to behave similarly to the $1/r$ behavior, and this indeed appears to be the case. Note that at $kd = 5$, the basic computational cell is nearly a wavelength long.

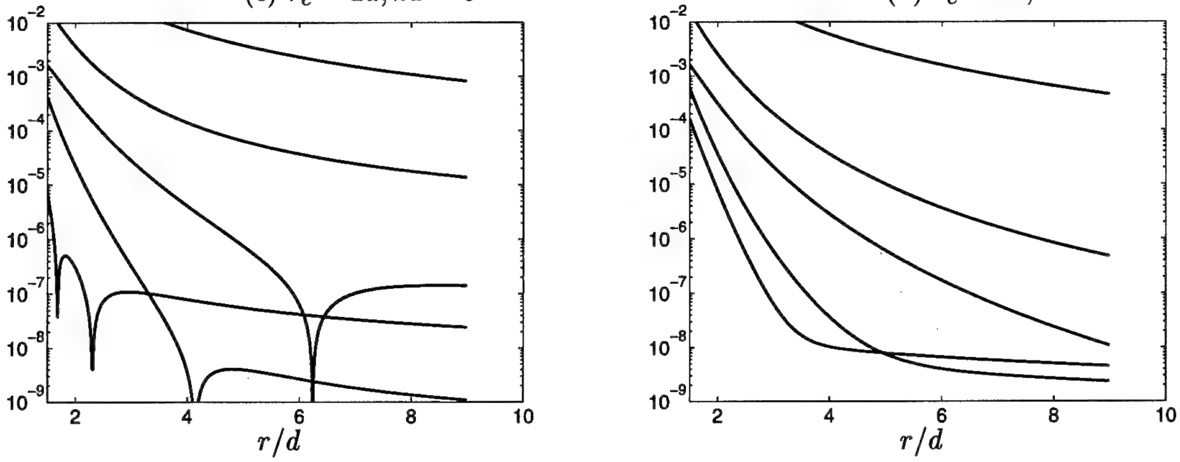
We also note that the collocation schemes are usually, but not always, more accurate than polynomial based projection. It is our experience in general that, at low order, the collocation-based schemes have, on average, errors 20%-40% smaller than polynomial-based approximations. The collocation-based schemes are particularly advantageous in terms of worst-case errors at high wavenumber, compare, e.g., Figure 2-11(a) and (f), where the collocation scheme can by superior be more than an order of magnitude.

Table 2-2 provides a final perspective on the projection schemes. It gives the errors for a point charge positioned as for the Figures 2-8 to 2-11, with evaluation points taken at the edge of a first-near-neighbor cube and second-near-neighbor cube. It is expected that these should be close to the worst-case approximation errors.



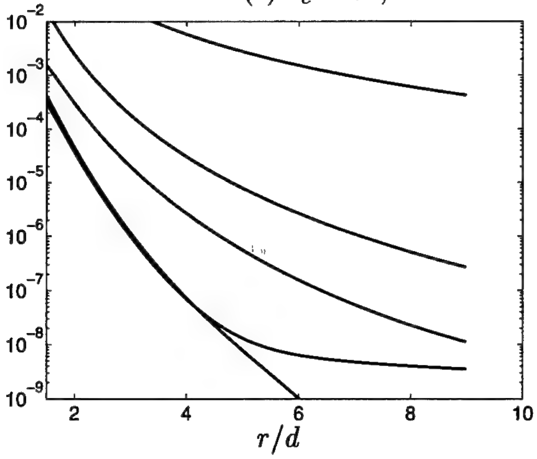
(a) polynomial, $kd = 0$

(b) $r_c = 1.5d, kd = 0$



(c) $r_c = 2d, kd = 0$

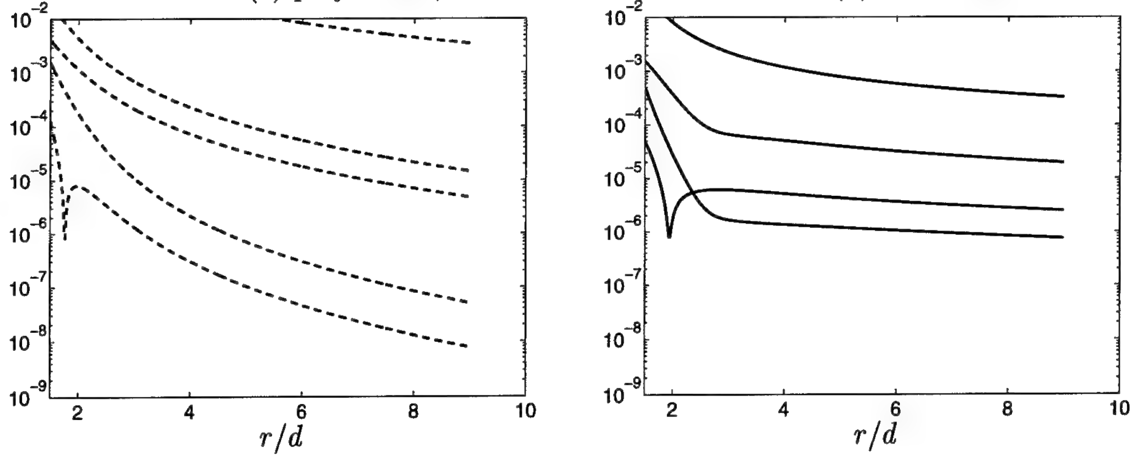
(d) $r_c = 5d, kd = 0$



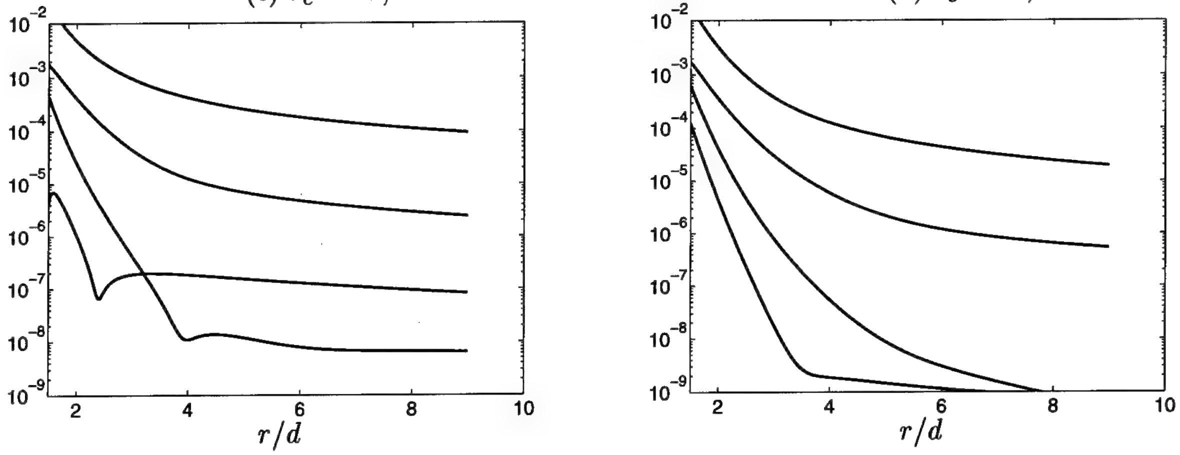
(e) $r_c = 8d, kd = 0$

FIGURE 2-8: Error in grid approximations. From bottom to top, $p = 6, 5, 4, 3, 2$.

(a) polynomial, $kd = 1$ (b) $r_c = 1.5d, kd = 1$



(c) $r_c = 2d, kd = 1$ (d) $r_c = 5d, kd = 1$



(e) $r_c = 8d, kd = 1$ (f) far-field $kd = 1$

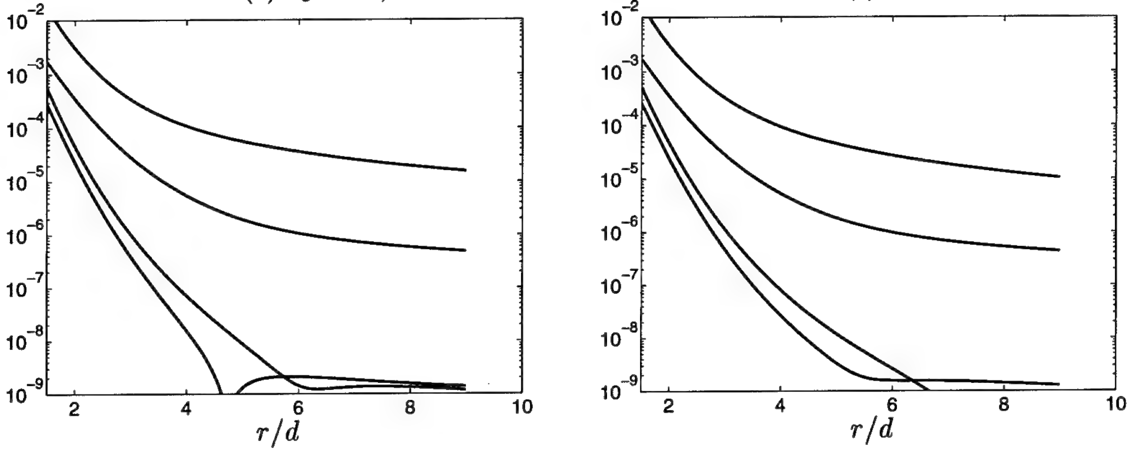
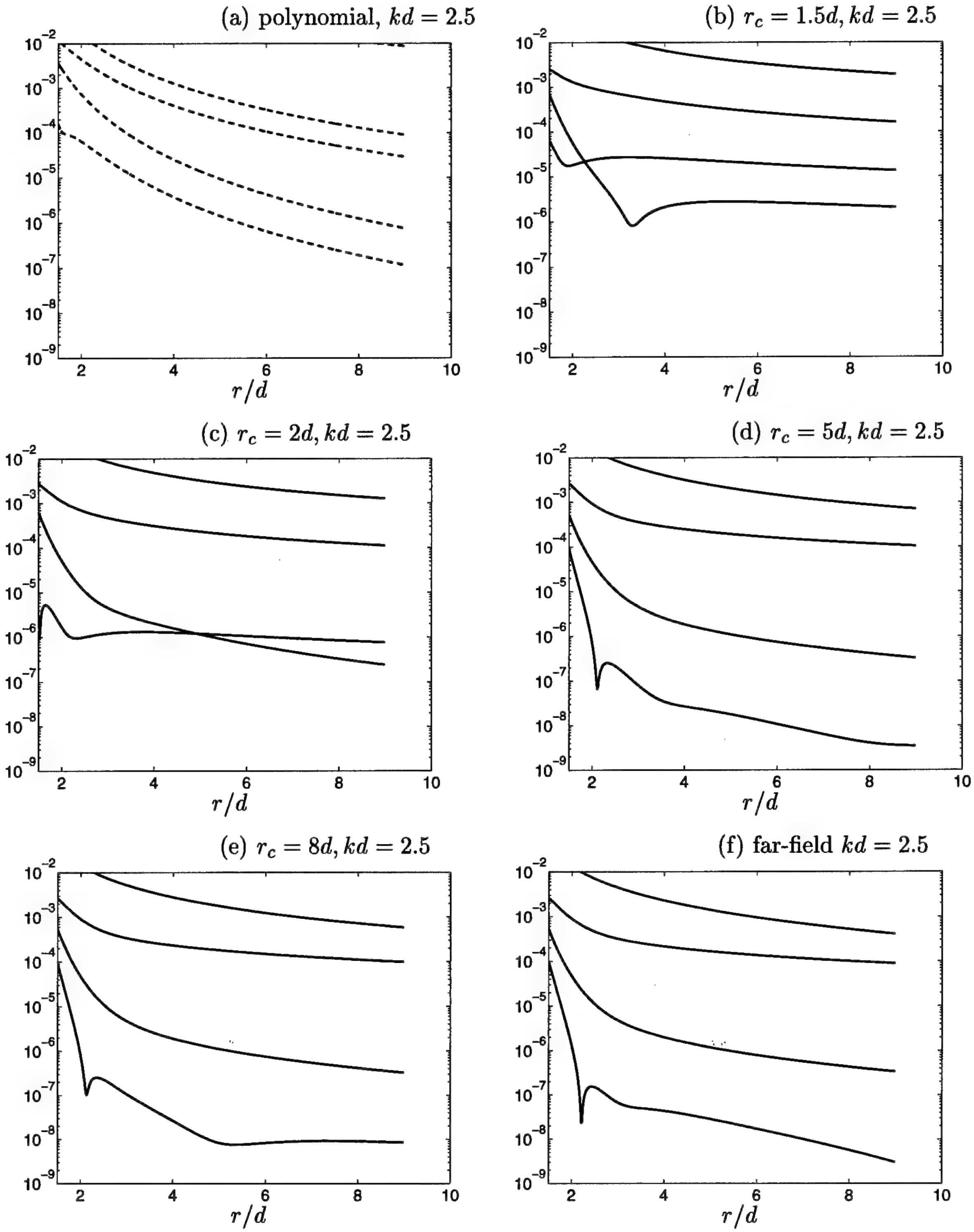


FIGURE 2-9: Error in grid approximations. From bottom to top, $p = 6, 5, 4, 3, 2$.


(a) polynomial, $kd = 2.5$

(b) $r_c = 1.5d, kd = 2.5$

(c) $r_c = 2d, kd = 2.5$

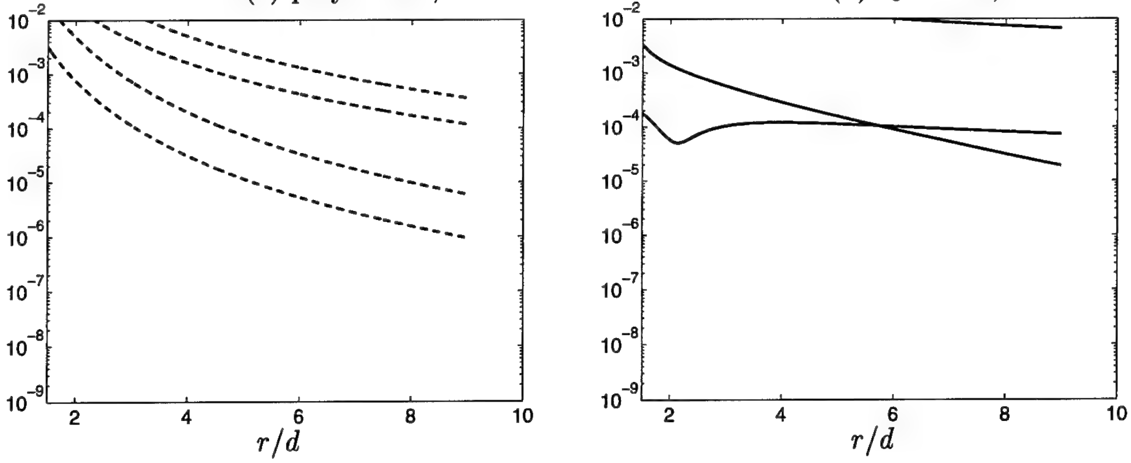
(d) $r_c = 5d, kd = 2.5$

(e) $r_c = 8d, kd = 2.5$

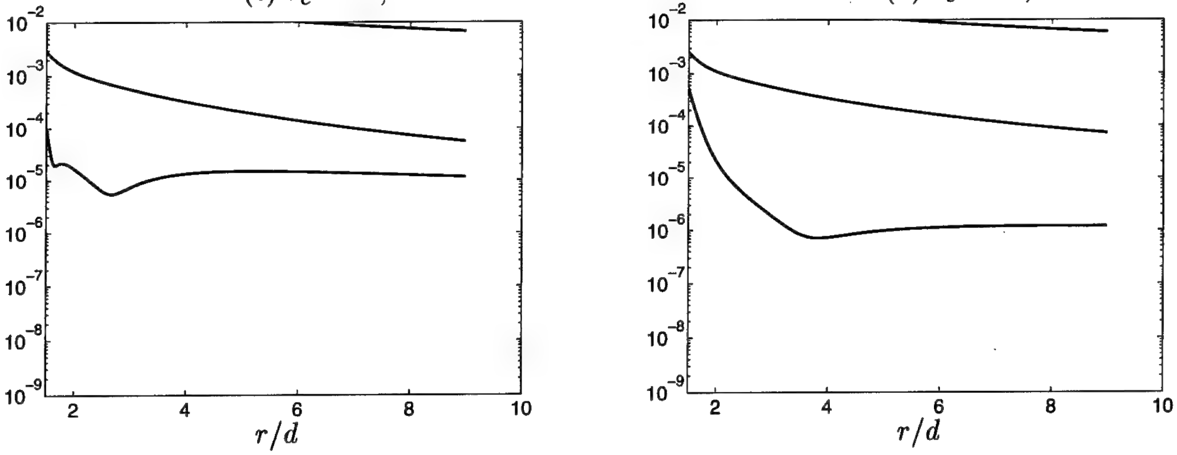
(f) far-field $kd = 2.5$

FIGURE 2-10: Error in grid approximations. From bottom to top, $p = 6, 5, 4, 3, 2$.

(a) polynomial, $kd = 5$ (b) $r_c = 1.5d, kd = 5$



(c) $r_c = 2d, kd = 5$ (d) $r_c = 5d, kd = 5$



(e) $r_c = 8d, kd = 5$ (f) far-field $kd = 5$

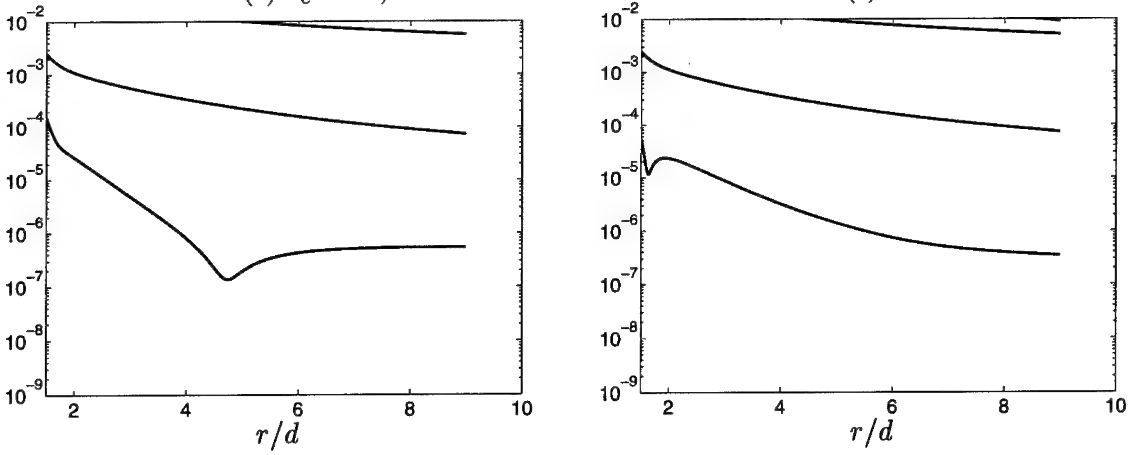


FIGURE 2-11: Error in grid approximations. From bottom to top, $p = 6, 5, 4, 3, 2$.

Method	kd	1st NN					2nd NN				
		$p=2$	$p=3$	$p=4$	$p=5$	$p=6$	$p=2$	$p=3$	$p=4$	$p=5$	$p=6$
Poly	0	0.18	0.017	0.0030	0.00136	0.00013	0.028	0.00072	0.00020	2.15e-05	1.82e-06
$r_c = 1.5d$	0	0.20	0.025	0.0014	0.00047	4.69e-05	0.036	0.00223	6.2e-05	1.75e-06	3.3e-06
$r_c = 2d$	0	0.19	0.019	0.0016	0.00041	6.64e-06	0.032	0.00112	8.79e-05	2.2e-06	1.09e-07
$r_c = 5d$	0	0.19	0.016	0.0016	0.00058	0.00016	0.029	0.00061	7.03e-05	3.65e-06	5.29e-07
$r_c = 8d$	0	0.18	0.016	0.0015	0.00041	0.00030	0.029	0.00057	6.8e-05	6.12e-06	5.21e-06
Poly	1	0.27	0.021	0.0040	0.00156	0.00013	0.065	0.00150	0.00044	3.57e-05	3.42e-06
$r_c = 1.5d$	1	0.27	0.028	0.0015	0.00050	4.89e-05	0.083	0.00366	0.00011	3.28e-06	6.13e-06
$r_c = 2d$	1	0.26	0.022	0.0018	0.00044	6.94e-06	0.075	0.00179	0.00012	2.69e-06	1.97e-07
$r_c = 5d$	1	0.24	0.018	0.0017	0.00062	0.00012	0.066	0.00090	8.95e-05	4.44e-06	2.23e-07
$r_c = 8d$	1	0.24	0.018	0.0017	0.00057	0.00028	0.065	0.00085	8.74e-05	5.62e-06	2.52e-06
far-field	1	0.24	0.018	0.0017	0.00053	0.00026	0.063	0.00082	8.64e-05	6.01e-06	2.83e-06
Poly	2.5	0.53	0.052	0.0118	0.00361	0.00016	0.15	0.0067	0.00202	0.00023	2.8e-05
$r_c = 1.5d$	2.5	0.98	0.065	0.0025	0.00070	6.3e-05	0.49	0.0165	0.00090	1.01e-05	2.74e-05
$r_c = 2d$	2.5	0.92	0.052	0.0027	0.00062	5.29e-06	0.46	0.0125	0.00065	1.06e-05	1.32e-06
$r_c = 5d$	2.5	0.84	0.042	0.0026	0.00052	8.82e-05	0.41	0.00865	0.00050	1.08e-05	2.17e-07
$r_c = 8d$	2.5	0.83	0.040	0.0026	0.00053	8.87e-05	0.40	0.00793	0.00048	1.1e-05	2.27e-07
far-field	2.5	0.80	0.039	0.0026	0.00052	9.74e-05	0.39	0.00693	0.00046	1.09e-05	1.49e-07
Poly	5	0.98	0.17	0.0454	0.0189	0.00318	0.29	0.0256	0.00785	0.00167	0.00026
$r_c = 1.5d$	5	3.54	0.55	0.0243	0.0034	0.00018	2.1	0.193	0.022	0.00084	0.00012
$r_c = 2d$	5	5.11	0.50	0.0237	0.0029	9.19e-05	2.9	0.175	0.0216	0.00077	1.49e-05
$r_c = 5d$	5	13.5	0.45	0.0204	0.0026	0.00053	7.2	0.151	0.0185	0.00074	4.84e-06
$r_c = 8d$	5	19.3	0.43	0.0196	0.0025	0.00015	10.2	0.144	0.0176	0.00075	1.13e-05
far-field	5	50.1	0.40	0.0185	0.0025	5.44e-05	28.7	0.132	0.0163	0.00077	1.49e-05

Table 2-2: Maximum errors at first- and second-near-neighbor cubes for various grid-projection schemes.

3

The Precorrected-FFT Algorithm

3.1 The Basic Idea

To develop a faster approach to computing the matrix-vector product, consider the parallelepiped which contains a three dimensional problem after it has been discretized into n panels. The parallelepiped containing the problem could be subdivided into an $k \times l \times m$ array of small cubes so that each small cube contains only a few panels. Figure 3-1(a) shows a discretized sphere, with the associated space subdivided into a $3 \times 3 \times 3$ array of cubes. These small cubes are be the *cells* referred to in the previous chapter.

The precorrected-FFT algorithm is motivated by the approximation scheme discussed in the previous chapter. Potentials at evaluation points distant from a cell can be accurately computed by representing the given cell's charge distribution using a small number of weighted point charges. If the point charges all lie on a uniform grid, for example at the cell vertices, then the computation of the potential at the grid points due to the grid charges is a discrete convolution which can be performed using the FFT. Figure 3-1(b) show a possible set of grid charge for the cell subvisions shown in Figure 3-1(a). Thus, a four-step method for approximating Pq is

1. project the panel charges onto a uniform grid of point charges,
2. compute the grid potentials due to grid charges using an FFT,
3. interpolate the grid potentials onto the panels, and
4. directly compute nearby interactions.

This process is summarized in Figure 3-2. We emphasize that the grid of point charges is introduced purely as a computational aid, it is not related to the underlying discretization of the conductors.

Given a set of M cells which contain the set of n panels and define the $\hat{n}$ grid points, we now describe how to compute the vector of potentials $\psi \in \mathbf{R}^n$ from the vector of panel charges

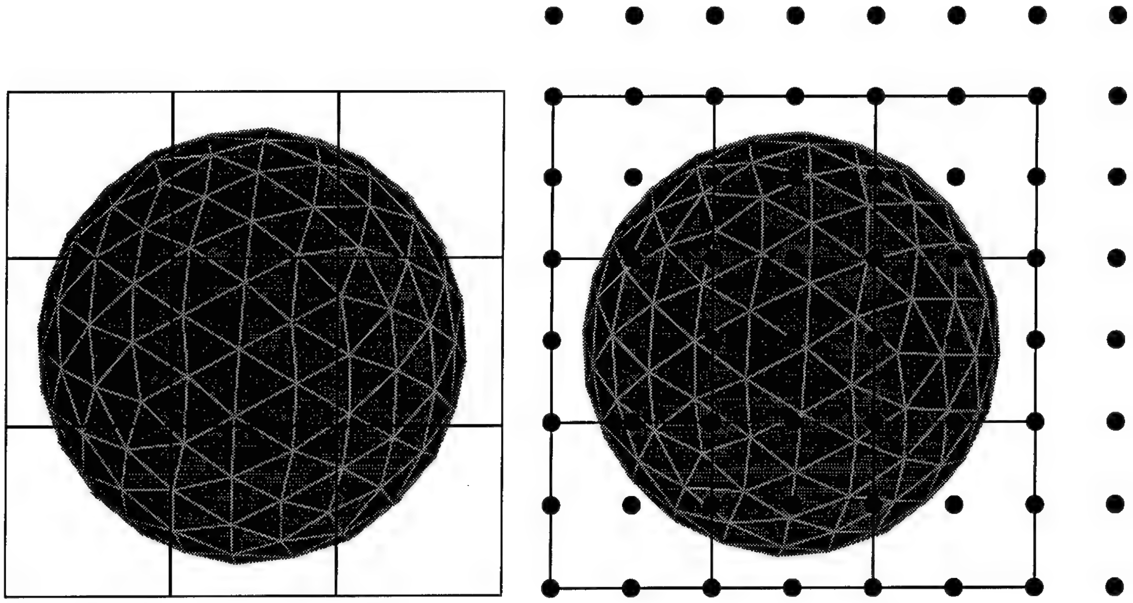


FIGURE 3-1: (a) Side view of a sphere discretized into 320 panels, with spatial decomposition into a $3 \times 3 \times 3$ array of cells. (b) Superimposed grid charges corresponding to the cell decomposition of (a), with $p = 3$. In each cell, a $3 \times 3 \times 3$ array of grid charges is used to represent the long range potential of the charged panels in the cell. Some of the grid charges are shared among cells. Note that the grid is "coarser" than the triangular panels used to discretize the sphere. The grid extends outside the problem domain because the number of grid points is required to be a factor of two.

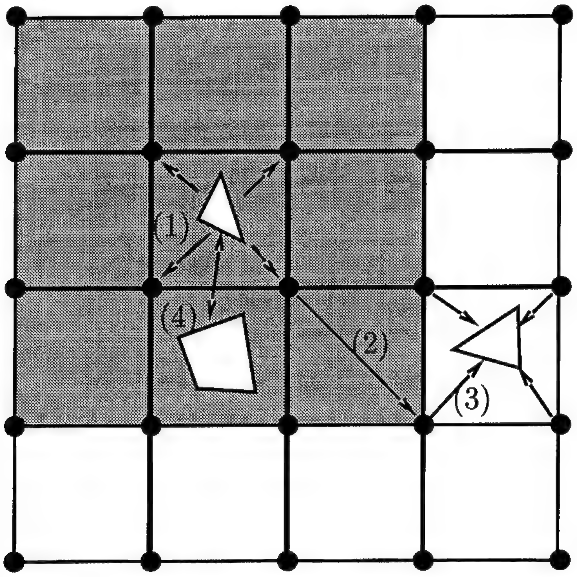


FIGURE 3-2: 2-D Pictorial representation of the four steps of the precorrected-FFT algorithm. Interactions with nearby panels (in the grey area) are computed directly, interactions between distant panels are computed using the grid.

$q \in \mathbf{R}^n$. $\psi_G \in \mathbf{R}^n$ will denote the contribution of the grid charges to the potentials on the n panel charges. $n(k)$ denotes the number of panels in a cell k, $q(k) \in \mathbf{R}^{n(k)}$ the restriction of the charge vector q to the indices whose corresponding panels lie in cell k and $\psi(k) \in \mathbf{R}^{n(k)}$ denotes the similar restriction of the potential vector. p denotes the order of grid approximation. $\hat{q} \in \mathbf{R}^{\hat{n}}$ is the vector of grid charges, $\hat{\psi}$ the vector of grid potentials, and $\hat{q}(k) \in \mathbf{R}^{p^3}$, $\hat{\psi}(k) \in \mathbf{R}^{p^3}$ denote the restriction of $\hat{q}$ and $\hat{\psi}$ respectively to grid points of cell k. We define $N(k)$ to be the indices of the set of cells which are "near" cell k. $W, V : \mathbf{R}^n \rightarrow \mathbf{R}^{\hat{n}}$, yet to be defined, will refer to linear operators which project n uniformly distributed panel charges to the $\hat{n}$ grid points, and the linear operator $H : \mathbf{R}^{\hat{n}} \rightarrow \mathbf{R}^{\hat{n}}$ gives grid potentials in terms of grid charges, i.e. $\hat{\psi} = H\hat{q}$. $W(k,j) : \mathbf{R} \rightarrow \mathbf{R}^{p^3}$ is the nonzero part of W corresponding to charge j in cell k, $1 \leq j \leq n(k)$; $V(k,j)$ is the similar part of V, and $H(k,l) : \mathbf{R}^{p^3} \rightarrow \mathbf{R}^{p^3}$ is the block of H which maps grid charges $\hat{q}(l)$ of cell l to grid potentials of cell k, $\hat{\psi}(k) = H(k,l)\hat{q}(l)$. A subscript indicates an index into a matrix or vector, e.g, $q_j(k)$ is the jth entry of vector $q(k)$.

3.2 Projecting onto a grid

The first step in the description of the algorithm is to describe the construction of the operator W. As was discussed in detail in Chapter 2, for panel charges contained within a neighborhood of a given cell, the potentials at evaluation points distant from the given cell can be accurately computed by representing the given cell's charge distribution with a small number of appropriately weighted point charges on a uniform grid thoughout the given cell's volume. Figure 3-1(b) shows the grid imposed on the cell structure of Figure 3-1(a) when a $3 \times 3 \times 3$ array of grid charges is used to represent the charge in each cell. Note that, because the grid is only used to represent the long-range part of the panel potentials, the grid may be significantly coarser than the actual problem discretization.

For any panel charge j in cell k, the projection operation generates a subset of the grid charges $\hat{q}(k)$. To obtain each panel's contribution to $\hat{q}$, it is necessary to calculate P^{qt} and $[P^{gt}]^{\dagger}$ Since P^{gt} is small and is the same for each cell, its singular value decomposition need only be performed once, and so the relative computational cost of calculating $[P^{gt}]^{\dagger}$ is insignificant. The final contribution to $\hat{q}(k)$ from the charges in cell k is generated by summing over all the charges in the cell. Note that panel charges outside cell k may contribute to some of the elements of $q(k)$ in the case of shared grid charges.

3.3 Computing Grid Potentials

Once the charge has been projected to a grid, the operation H, computing the potentials at the grid points due to the grid charges, is a three-dimensional convolution. We denote this as

$$\hat{\psi}(i,j,k) = H\hat{q} \equiv \sum_{i',j',k'} h(i-i',j-j',k-k')\hat{q}(i',j',k') \tag{3.1}$$

where i,j,k and i',j',k' are triplets specifying the grid points and $h(i-i',j-j',k-k')$ is the inverse distance between grid points i,j,k and i',j',k'. As will be made clear below, $h(0,0,0)$ can be arbitrarily defined, and is set to zero. The above convolution can be rapidly computed by using the Fast Fourier Transform (FFT). In practice, each convolution requires one forward and one inverse three-dimensional FFT. The discrete Fourier transform of the kernel matrix H, denoted $\tilde{\mathbf{H}}$, need be computed only once.

An efficient FFT implementation is central to the performance of the precorrected-FFT algorithm. The FFT is a very well-studied algorithm, and many possible implementations exist. Most FFT implementations have a fairly regular nature, and thus very efficient optimized code can be developed. Also, the structure of the data in a multidimensional convolution can be exploited for additional performance gains. For example, the use of the FFT to perform a linear multidimensional convolution involves embedding the data ($\hat{q}$) to be transformed into a larger data space, much of which is zero. The fact that much of the transformed data is zero can be exploited to yield a more efficient transform. In comparison, achieving optimal machine performance with fast multipole algorithms is more difficult, due to the less regular nature of the algorithms.

3.4 Interpolating Grid Potentials

Once the grid potentials have been computed, they must be interpolated to the panels in each cell. When a collocation scheme is used to discretize the integral equation, the operator which interpolates potential at grid points in cell k to a charge j also in cell k is not $[W(k,j)]^T$ defined in Eq. 2.19. Instead, the projection operator $V(k,j)$ for a point charge located at the collocation point is computed which gives the interpolation operator $[V(k,j)]^T$. However, if a Galerkin scheme is used in the discretization then the interpolation operator is $[W(k,j)]^T$.

Thus, projection, followed by convolution, followed by interpolation gives the grid-charge approximation ψ_G to the potentials which can be represented as

$$\psi_G = V^T HWq. \tag{3.2}$$

If Galerkin methods are used, Eq. 3.2 becomes

$$\psi_G = W^T HWq \tag{3.3}$$

and therefore the precorrected-FFT method preserves the symmetry of the Galerkin discretization for free-space problems.

3.5 Precorrecting

The difficulty with the above three steps is that the calculations using the FFT on the grid do not accurately approximate the nearby interactions. In $\hat{\psi}$ of (3.2), the portions of Pq associated with neighboring cell interactions have already been computed, though this close interaction has been poorly approximated in the projection/interpolation. A more accurate calculation of interactions between nearby panels is needed, but it is also necessary to remove or avoid the inaccurate contribution from the use of the grid. This is a general difficulty with grid-based potential calculation methods, and a variety of correction methods have been proposed[44, 12, 6], the details of which usually depend on the problem being solved, the interpolation scheme, and the nature of the grid solver.

Because our algorithm works directly with the Green function, and because the iterative solver requires that many potential evaluations are performed for a given panel configuration, it is possible to treat nearby panel interactions exactly, without sacrificing algorithmic efficiency. We accurately treat interactions between panels close together by modifying the way nearby interactions are computed, a step we refer to as precorrection.

In particular, denote as $P(k,l)$ the portion of P associated with the interaction between neighboring cells k and l, $V(k)$ and $W(l)$ the matrices formed from the columns $V(k,j)$ and $W(k,j)$ respectively, and denote $\psi_{(k|l)}$ as the panel potentials in cell k due to the charges q_l in cell l. Then

$$\psi_{G,(k|l)} = V(k)^T H(k,l) W(l) q_l \tag{3.4}$$

is the grid-approximation to $\psi_{(k|l)}$, which is inaccurate. Subtracting this approximation and then adding the correct contribution,

$$\psi_{(k|l)} = \psi_{G,(k|l)} + \left(P_{k,l} - V(k)^T H_{k,l} W(l)\right) q_l, \tag{3.5}$$

produced the accurate result $P(k,l)q_l$.

This may be efficiently accomplished by defining

$$\tilde{P}(k,l) \equiv P(k,l) - V(k)^T H(k,l) W(l) \tag{3.6}$$

to be the "precorrected" direct interaction operator. When used in conjunction with the grid charge representation $\tilde{P}(k,l)$ results in exact calculation of the interactions of panels which are close. Assuming that the Pq product will be computed many times in the inner loop of an iterative algorithm, $\tilde{P}$ will be expensive to initially compute, but will cost no more to subsequently apply than P.

3.6 Complete Algorithm

Combining the above steps leads to the precorrected-FFT algorithm, which rapidly computes the Pq dense matrix-vector product. Using the above notation, the algorithm can be described as two steps. The first step is to compute

$$\psi_G = V^T H W q. \tag{3.7}$$

W and V are sparse interpolation operators, and H can be represented in a sparse manner via the FFT. The second step is to add in the corrected direct interactions, to obtain the panel potentials $\psi(k)$ for each cell k,

$$\psi(k) = \psi_G(k) + \sum_{l \in N(k)} \tilde{P}(k,l) q_l. \tag{3.8}$$

Because for each k, $N(k)$ is a small set and each matrix $\tilde{P}(k,l)$ is also small, this second step is also a sparse operation. The complete algorithm is given below in pseudocode form.

```
Precorrected-FFT Algorithm to Compute $Pq$
  /* Projection Step */
    Set $\hat{q} = 0$
    For each cell $k = 1$ to $M$ {
      For each panel $j$ in cell $k$, $j = 1$ to $n(k)$ {
        Add $\hat{q}(k) = \hat{q}(k) + W(k,j)q_j(k)$
      }
    }
  /* Convolution Step */
    Compute $\hat{\mathbf{Q}} = \mathrm{FFT}(\hat{q})$
    Compute $\hat{\mathbf{\Psi}} = \tilde{\mathbf{H}}\hat{\mathbf{Q}}$
    Compute $\hat{\psi} = \mathrm{FFT}^{-1}(\hat{\mathbf{\Psi}})$
  /* Interpolation Step */
    Set $\psi = 0$
    For each cell $k = 1$ to $M$ {
      For each panel $j$ in cell $k$, $j = 1$ to $n(k)$ {
        Add $\psi_j(k) = \psi_j(k) + [V(k,j)]^T \hat{\psi}_j(k)$
      }
    }
  /* Nearby Interactions */
    For each cell $k = 1$ to $M$ {
      For each cell $l$ in $N(k)$
        $\psi(k) = \psi(k) + \tilde{P}(k,l)q(l)$
      }
    }
```

Thus, the effect of this algorithm is to replace the operation

$$\psi \leftarrow Pq, \tag{3.9}$$

where P is a dense matrix, with the operation

$$\psi \leftarrow \left[\tilde{P} + V^T H W\right] q \tag{3.10}$$

where all the matrices $\tilde{P}, V, H, W$ possess sparse representations.

3.7 Grid selection

Before the algorithm has been completely specified, it it necessary to specify how panels are selected for inclusion in direct interaction regions and how the grid size is selected. That is, for each cell k the set $N(k)$ must be specified. To insure that interactions between panels which are close together are treated accurately, at a minimum it is necessary to compute interactions between panels in cells which are near-neighbors of each other via direct products. The near-neighbors of a cell β are defined to be all the cells which have a vertex in common with cell β (thus a cell is a near-neighbor of itself). We have included only near-neighbor interactions in the computations of this paper.

The worst-case accuracy of the grid representation is a function of the ratio of the cell radius to the radius of the direct interaction region[38]. Thus, once the direct-interaction region has been specified to be near-neighbor cells, the selection of the cell size, and hence the grid spacing is purely a matter of computational efficiency. The cost of direct interactions will decrease monotonically as the cells are made smaller, but the number of grid points will increase, raising the cost of the FFT. This implies that the total cost of the algorithm will have a minimum for some grid spacing. For a given grid spacing and panel configuration, the memory and computation time needed by precorrected-FFT algorithm can be estimated cheaply, so the optimal grid spacing can be obtained by starting with a small number of grid points and increasing the number until a minimum CPU or memory estimate, as appropriate, is reached. In addition, we have generally required that the number of grid points be a factor of two, in order to exploit efficient FFT implementations.

It is interesting that the optimal grid size may occasionally be such that the number of grid charges $\hat{n}$ is *larger* than the original number of panel charges n. This may be the case even when the grid spacing is larger than the underlying panel sizes, that is, when the grid is "coarser" than the panel discretization. Such a case may occur, for example, for a finely discretized cube surface, where the grid must fill the three-dimensional space of the cube's interior. However, the overall algorithm may still be quite effective, since the cost of the FFT is $O(\hat{n} \log \hat{n})$, with a constant factor of $O(10)$. Thus if $\hat{n} \simeq n$ and n is large, the cost of the FFT is less than that of the direct product by a factor of nearly $O(n)$, and so the algorithm may have $\hat{n} > n$ by a fairly significant factor and still possess an advantage over the direct computation.

Preceding Page Blank

4

Complexity Analysis

In this chapter, we analyze the complexity of the precorrected-FFT algorithm and give some comparisons with other approaches. It is shown that for homogeneous problems, the method is order $n \log n$ nearly independent of the kernel. For an inhomogeneity generated by a very finely discretized surface, the combined method slows to order $n^{6/5}$ for $1/r$ kernels (or any discretizations which are geometrically restricted) and order $n^{4/3}$ for Helmholtz kernels (discretizations which are restricted by wavelength).

First we compare the efficiency of the grid representation used in the precorrected-FFT algorithm to the multipole expansions used in the fast multipole method.

Both the fast multipole algorithm and the precorrected-FFT algorithm obtain efficiency by representing the long-range part of the potential of a group of charges by an expression which can be used at multiple evaluation points, but the algorithms differ in the way they cluster sets of charges together to form single expressions.

Again consider subdividing the parallelepiped containing the entire three-dimensional problem domain into a $k \times l \times m$ array of cells. Then, the collocation approach above can be used to generate point charge approximations for charge distributions in every cell, effectively projecting the charge density onto a three-dimensional grid. For example, if the representative point charges are placed at the cell vertices, then the panel charge distribution will be projected to a $(k+1) \times (l+1) \times (m+1)$ uniform grid. Fast multipole algorithms also effectively create a uniform grid by constructing multipole expansions at the center of each cell, but due to sharing, the point charge approach can be more efficient. For example, when representing the potential of a panel by charges at the cell vertices, there are eight free coefficients which may be varied to obtain an optimal representation, and there will be $(k+1)(l+1)(m+1)$ terms in the entire domain. On average, there is only one grid-charge per cubic cell, since a point charge at a cell vertex is used to represent charge in the eight cells which share that vertex. By contrast, as no sharing occurs in the the multipole representation, if there are q coefficients in the multipole expansion which represents the potential of the charges in the cell, the total number of terms in the domain will be $q(k+1)(l+1)(m+1)$. For an equivalent number of total terms in the domain, we expect the grid representation to be more accurate. Conversely, for roughly equivalent accuracy, we may choose $q \simeq 8$, but then the total number of multipole terms will be significantly higher than for the grid representation.

From the analysis of the preceding paragraph, we expect the grid representation to be locally more efficient than the use of multipole expansions. However, our current implementation of the precorrected-FFT algorithm may be globally less efficient, as the grid representation is introduced throughout space, even where no panels are present. Thus, whereas for a problem containing n panels, the fast multipole algorithm can perform a potential evaluation for all of the panels in $O(n)$ operations, regardless of the panel distribution[16], no such guarantee is available for the precorrected-FFT algorithm. However, it is possible to establish a weaker complexity result for the precorrected-FFT method:

Theorem 4.1. For a homogeneous distribution of n panels, the precorrected-FFT method requires $O(n \log n)$ operations to perform a potential calculation.

Proof. Given that the computational domain is a parallelepiped containing n panels, again assume space has been divided into an array of $k \times l \times m$ cells, and that the panel distribution is homogeneous on the scale of the cell size. That is, the number of panels per cell, N_c, is bounded independent of n, with klm of order-n. Finally, assume that the grid in each cell is a $p \times p \times p$ array. There are three components in the cost of the precorrected-FFT algorithm: the cost of direct interactions, the cost of grid projection and interpolation, and the cost of the FFT. The cost of the direct interactions will be $O(N_c^2 \times klm) = O(klm) = O(n)$. The cost of the grid projection will be $O(np^3) = O(n)$. Finally, the cost of the FFT will be $O(p^3 klm \log p^3 klm) = O(n \log n)$. Summing these costs results in the final complexity of $O(n \log n)$.

□

Since the grid spacing is typically less fine than the underlying surface discretization, for a typical problem the precorrected-FFT algorithm has $O(n \log n)$ complexity for problems with considerable inhomogeneity in the fine surface discretization, as long as the panel distribution is homogeneous at a very coarse level. As will be seen in Section 5.3, many structures arising in practice satisfy this "coarsely homogeneous" condition.

For the boundary-integral methods considered in this paper, however, the panels are usually not homogeneously distributed.

Definition 1. Suppose S is the boundary of a simply-connected volume in $\mathbf{R}^3$ oriented about its principle axes. If $\{\Delta\}$ is the set of panel diameters, $\{\Delta\}$ is an homogeneous partition of S if there exists integers $(\overline{k}, \overline{l}, \overline{m})$ such that

$$\inf \Delta > \frac{L_1}{\overline{k}}, \inf \Delta > \frac{L_2}{\overline{l}}, \inf \Delta > \frac{L_3}{\overline{m}} \tag{4.1}$$

and

$$\sup \Delta < c\frac{L_1}{\overline{k}}, \sup \Delta < c\frac{L_2}{\overline{l}}, \sup \Delta < c\frac{L_3}{\overline{m}} \tag{4.2}$$

for some c independent of Δ.

Theorem 4.2. For a homogenous partition of a single closed surface S, at fixed k the precorrected-FFT method requires $O(n^{6/5}\log n)$ *operations to perform a potential calculation, where n is the number of panels.*

Proof. Again assume space has been divided into an array of $k \times l \times m$ cells. From the homogeneous partition condition we can conclude that there exists constants c_1, c_2 such that

$$c_1(\overline{k}\,\overline{l} + \overline{k}\overline{m} + \overline{l}\overline{m}) < n < c_2(\overline{k}\,\overline{l} + \overline{k}\overline{m} + \overline{l}\overline{m}) \tag{4.3}$$

The cost of the direct interactions is

$$C_D = O\left(\frac{(\overline{k}\,\overline{l})^2}{kl} + \frac{(\overline{k}\overline{m})^2}{km} + \frac{(\overline{l}\overline{m})^2}{lm}\right) \tag{4.4}$$

and the cost of the FFT is

$$C_F = O(klm \log klm) \tag{4.5}$$

The cost of the projection and interpolation steps is neglected as they are both always $O(n)$ in cost. Suppose for the moment that $k = l = m, \overline{k} = \overline{l} = \overline{m}$, and for simplicity neglecting the logarithmic factor, then $n = O(\overline{k}^2)$, $C_D = O(\overline{k}^4/k^2), C_F = O(k^3)$. Optimizing $C_D + C_F$ for k gives $k = O(\overline{k}^{4/5})$ and the costs $C_D = O(n^{6/5}), C_F = O(n^{6/5})$. This motivates us to choose

$$k = O(\overline{k}^{4/5}), l = O(\overline{l}^{4/5}), m = O(\overline{m}^{4/5}). \tag{4.6}$$

In that case,

$$C_F = O\left(\overline{k}\,\overline{l}\overline{m}\right)^{4/5} \log \overline{k}\,\overline{l}\overline{m} < c_3 n^{6/5} \log n \tag{4.7}$$

and

$$C_D = O\left((\overline{k}\,\overline{l})^{6/5} + (\overline{k}\overline{m})^{6/5} + (\overline{l}\overline{m})^{6/5}\right) < c_4 n^{6/5} \tag{4.8}$$

for constants c_3, c_4. Thus the cost is bounded by $O(n^{6/5}\log n)$. □

In this analysis, we have assumed that p is constant. For a given problem, when solving the Helmholtz discretization as the frequency increases, generally the number of panels must increase to retain a fixed number of panels per wavelength. However, the size of a computational cell decreases proportional to $1/M$, or as $n^{-4/5}$, slower than n. Thus, for high frequencies the criterion in (2.48) that the order of the quadrature rule be greater than $2k\Delta$ will be violated. We must allow p to vary with n to obtain the correct complexity analysis, which gives a different complexity bound.

Theorem 4.3. For a single closed surface the precorrected-FFT method with $\sqrt{n}$ inversely proportional to the wavelength λ[1] *requires at most $O(n^{4/3}\log n)$ operations to perform a potential calculation, where n is the number of panels.*

Proof. The analysis is similar to the proceeding theorem. Assume that $\bar{l} = \bar{k} = \overline{m}$, so $k = l = m$ and $n = O(\overline{m}^2)$. Assume the size L of the computational domain is fixed. Generally, a fixed number of panels per wavelength required to maintain the solution accuracy, i.e. $\overline{m} \sim L/\lambda$. Theorem 2.5 implies that it is necessary to require that the order q of the collocation rule must be

$$q \sim \frac{\Delta}{\lambda}, \tag{4.9}$$

so

$$q \sim \frac{L}{\lambda m} \sim \frac{\overline{m}}{m} \tag{4.10}$$

The number of collocation points necessary for order q quadrature is $O(q^2)$. *If we assume that it is possible to satisfy the collocation equations with $p^3 \sim l^2$ grid charges*, then we have

$$p \sim (\frac{\overline{m}}{m})^{2/3}. \tag{4.11}$$

Repeating the above complexity analysis, we have

- Direct cost $C_D = O(\overline{m}^4/m^2)$
- Interpolation cost $C_I = O(p^3\overline{m}^2) = O(\overline{m}^4/m^2)$, same order as the direct cost
- FFT cost $C_F = O(m^3p^3\log_2 mp) = O(m\overline{m}^2\log m\overline{m}^2)$

Neglecting the log factor, The total cost is thus $C_T = O(m\overline{m}^2 + \overline{m}^4/m^2)$ which when optimized for m gives

$$m = O(\overline{m}^{2/3}) \tag{4.12}$$

The asymptotic cost of the entire algorithm is then $O(n^{4/3}\log n)$, a slight increase over the $O(n^{6/5}\log n)$ in the case of Poisson's equation, and competitive with two-level multipole based schemes for the Helmholtz equation [18].

We should also note that the cost of forming the grid projection operators, $O(p^9) = O(\overline{m}^2) = O(n)$ remains reasonable. □

Three comments are in order.

First, The critical assumption in the above proof is that it is possible to satisfy the collocation equations with $p^3 \sim l^2$. Certainly this is true if the grid-potential mapping matrix has full rank, which is the case for low to moderate frequency problems. It has been experimentally verified that $p^3 \sim l^2$ condition is satisfiable at least up to $p = 10$, which is sufficient so solve problems with the cell size Δ on the order of a wavelength, which is a very high frequency problem. For such high frequency problems other algorithms are likely to be more suitable in any case.

---

[1] This is the standard fixed-number-of-panels per wavelength criterion.

Second, For many engineering problems of interest, the discretization is geometry-limited, not frequency limited, in which case the complexity estimate for the precorrected-FFT method applied to surfaces is $O(n^{6/5} \log n)$.

Third, if polynomials are used to generate the projection/interpolation operators, the complexity for a geometrically-limited discretization refinement (the analogous theorem is 4.2), the complexity remains $O(n^{6/5} \log n)$, though in general the constant factor will increase due to the less accurate approximations. For wavelength-limited discretizations, the complexity becomes $O(n^{3/2} \log n)$. Volume-filling geometries are still solved at a cost of $O(n \log n)$ in both cases.

It is instructive to consider these results in mind of other available grid-based algorithms. In order to solve the underlying potential-theoretic problem, the precorrected-FFT algorithm introduces a uniform grid which covers the problem domain volume. Some comments are in order on the differences and similarities between the precorrected-FFT algorithm and other methods which introduce volumetric grids.

First, most other methods which use a grid to represent the solution throughout space, such as finite-difference methods, finite-element methods, or integral equation methods which directly exploit the convolutional properties of the kernel via the FFT[45, 46, 47], introduce a space-filling grid which must also accurately represent the complicated problem geometry. These two conflicting requirements generally result in either restricted geometries or a very large number of unknowns that in turn limits the size of problem that can be effectively solved.

In contrast, as shown in Figure 3-1, the grid introduced by the precorrected-FFT algorithm is geometrically unrelated to the underlying surface discretization of the geometry. In general the number of panels in a surface discretization is much smaller than the number of elements in a volume representation, so we expect the precorrected-FFT algorithm to be more efficient, than, for example, finite-difference approaches.

Additionally, most other three-dimensional grid-based approaches necessarily have a complexity of k^3, if k is the number of basis elements along a side. The precorrected-FFT method analyzed here uses $O(n \simeq k^2)$ basis elements in the underlying surface discretization, and the complexity is $O(n) \rightarrow O(n^{1.2}) = O(k^2) \rightarrow O(k^{2.4})$. At $k = 50$ basis elements per dimension, corresponding only to a 15,000 panel problem, $k^{3.0}$ exceeds $k^{2.4}$ by more than a factor of 20.

In short, because of the decoupling of short-range interactions from the long-range interactions treated by the grid, the precorrected-FFT method can efficiently utilize fast potential solvers without sacrificing the ability to represent complicated surface geometries in a compact manner.

5

Algorithm Performance on Laplace Problems

Many diverse engineering applications, such as analysis of signal integrity in integrated-circuit interconnect, characterization of electrical packaging, and design of microelectromechanical systems[48] require rapid, accurate electrostatic analysis of complicated three-dimensional structures. Often the electrostatic analysis requires the solution of the Laplace equation with Dirichlet or, less often, Neumann boundary conditions specified on a complicated surface that lies in three dimensions.

This problem can be solved as discussed in Chapter 1: by formulating a first-kind integral equation with $1/r$ kernel which is then discretized by a weighted-residuals technique. Other recent work has been based on random-walk methods[49], partitioning heuristics combined with techniques from matrix extension theory[50, 51], finite-difference[52, 53] or finite-element methods[54, 55].

In this chapter the effectiveness of the precorrected-FFT approach in performing the matrix-vector product needed in an iterative solution of the dense discretized system, such Equation 1.5, is demonstrated via several empirical studies. First we present a study of the global errors introduced by the method. Then, as a prototype application, we consider the extraction of coupling capacitances in three-dimensional geometries. We present extensive experimental comparisons with the capacitance extraction code FASTCAP[7] and demonstrate that, for a wide variety of geometries commonly encountered in VLSI design , the precorrected-FFT algorithm is superior to the fast multipole algorithm used in FASTCAP in terms of execution time and memory use. At engineering accuracies, in terms of a speed-memory product, the new algorithm can be superior to the fast multipole based schemes by more than an order of magnitude.

5.1 Empirical Error Analysis

In this section we examine some simple examples in order to evaluate the errors introduced by the grid projection method.

As described above, in the precorrected-FFT algorithm, the interaction between panels in neighboring cells is computed exactly, but more distant interactions are approximated by projection, convolution, and then interpolation using the grid. To demonstrate that the errors due to using the grid are well-controlled, we present an empirical error study based on an analytically solvable potential problem borrowed from [16]. If Equation 1.2 is solved on a sphere with given potential

$$\psi(\theta,\phi) = -\frac{\cos\theta}{2} \tag{5.1}$$

the analytically computable charge distribution is

$$\sigma(\theta,\phi) = -\frac{3\cos\theta}{8\pi}\,. \tag{5.2}$$

To estimate the error introduced by the grid approximations in the precorrected-FFT method, the sphere can be discretized, as in Figure 5-1, and the charges q_i on each panel i computed. The approximations introduced by the grid-charge approximation to long-range interations will become evident as the discretization is refined, since eventually these errors will dominate over the discretization error. One relative measure of the error is

$$\varepsilon = \frac{1}{Q}\sum_i |q_i - a_i\sigma(x_i)| \tag{5.3}$$

where the sum runs over all panels i, x_i is the centroid of panel i, a_i the area of that panel, q_i the charge on the panel, and Q the exact total charge on the sphere. Figure 5-1 (b) shows that for the low-order piecewise-constant collocation scheme, as the discretization of the sphere is refined (see Fig. 5-1 (a)), the integrated error ε decreases proportional to $1/n$. The multipole or precorrected-FFT approximation errors are evident when ε ceases to decrease as n is increased. For example, for the $2 \times 2 \times 2$ grid-charge representation is used in each cell ($p = 2$), ε ceases to decrease below about 0.05. This indicates that the $p = 2$ grid charge scheme introduces errors into the integrated charge calculation of about 5%. Similarly, we expect the $p = 3$ scheme to be accurate to almost a tenth of a percent. We have also shown results for the $l = 2$ multipole approximation, which from this experiment we expect to be intermediate in accuracy between the grid $p = 3$ and $p = 2$ approximations. Note that these errors are smaller in a relative sense than the worst-case errors presented in Table 2-2.

5.2 Reference Examples

In this section we examine a variety of simple examples to evaluate the performance of the precorrected-FFT algorithm and illustrate some of its shortcomings.

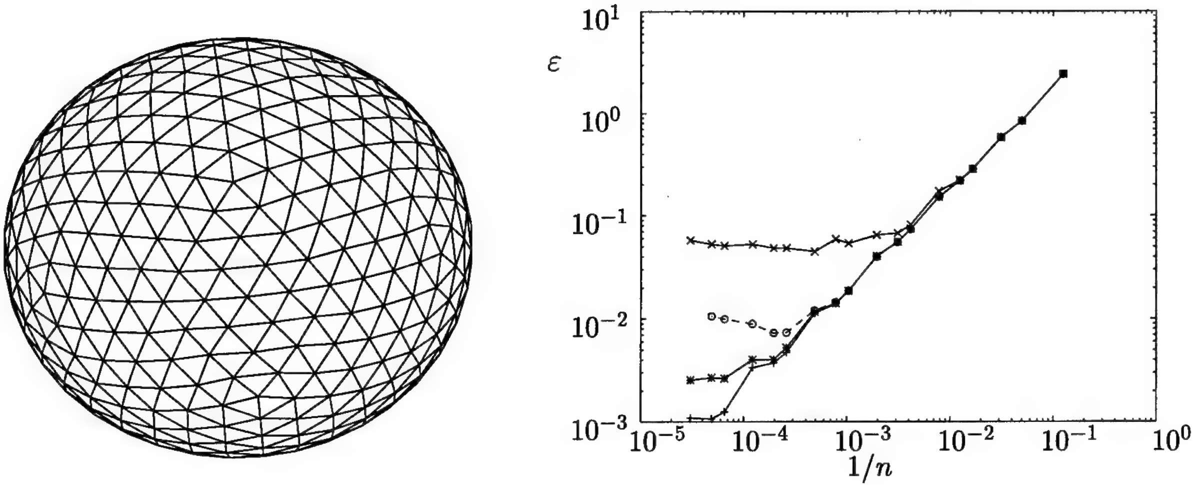


FIGURE 5-1: (a) A sphere discretized into 960 panels. The discretization is refined by subdividing the spherical triangle defined by the panel vertices into four triangular panels, whose vertices are the midpoints of the edges of the original spherical triangle. (b) Solid lines show integrated charge error for the sphere with Dirichlet condition of Eq. 17. Solid line shows errors for grid code, (x) $p = 2$ (*) $p = 3$ (+) $p = 4$. Dashed line connecting (o) shows error for $l = 2$ multipole scheme.

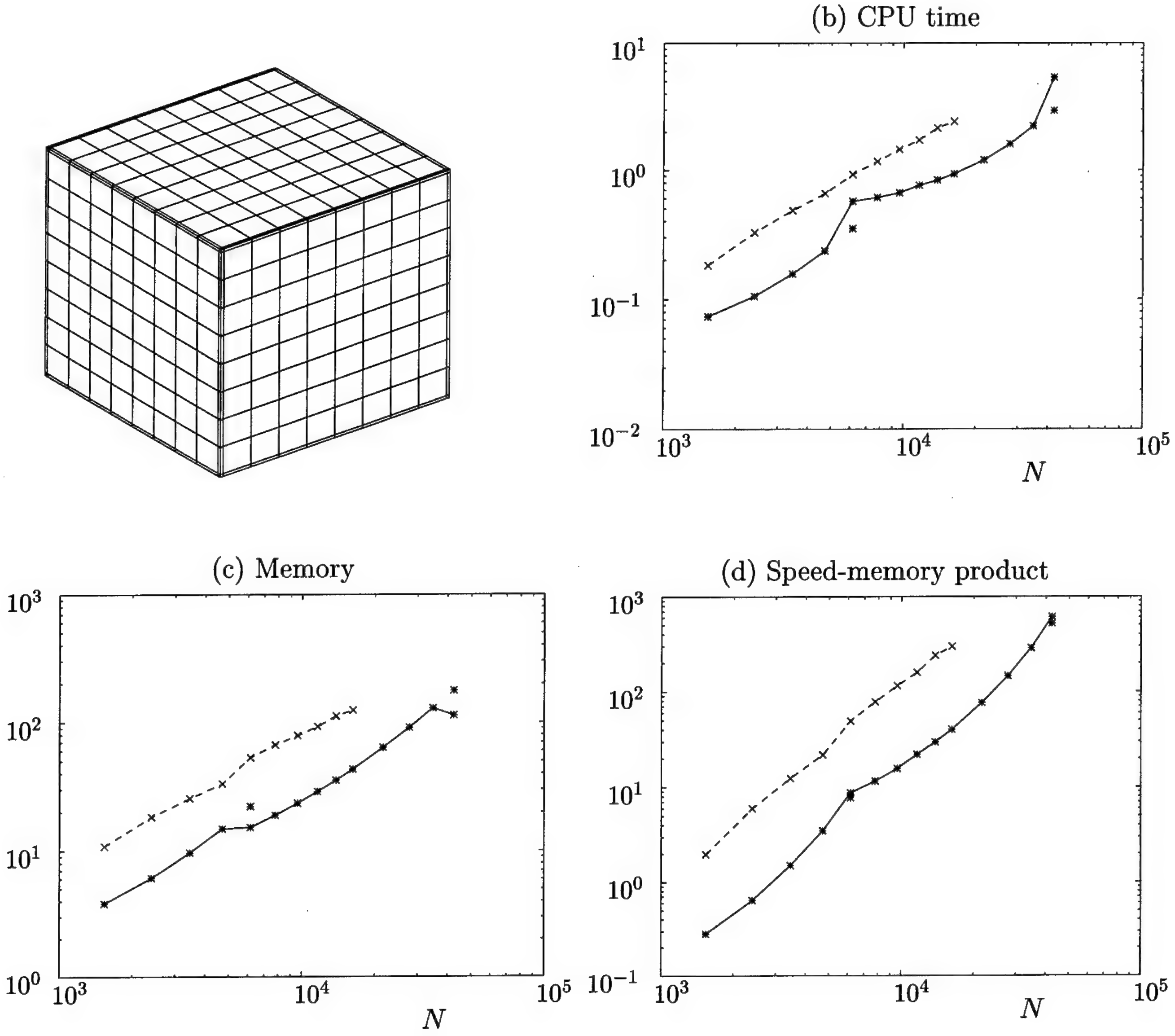


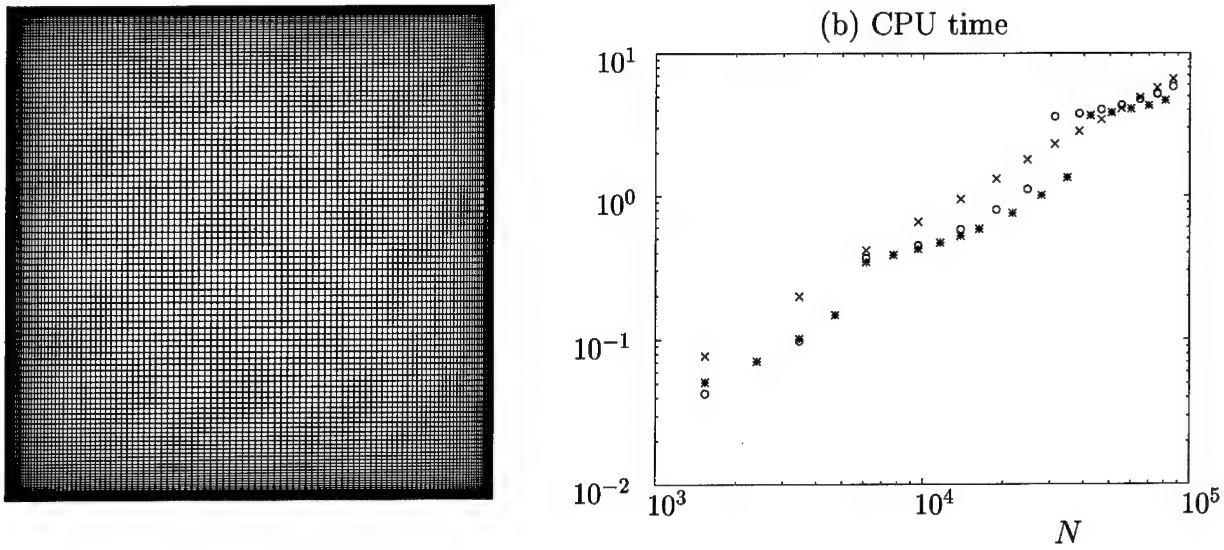
FIGURE 5-2: The cube example. (a) Discretization of the cube. (b) CPU time, in seconds, needed for the fast multipole (dashed line connecting 'x') and precorrected-FFT algorithms (*) to compute a matrix-vector product. For the precorrected-FFT algorithm, different results are possible depending on whether speed or memory usage is to be optimized. The solid line connects runs with grid sizes chosen to minimize memory use. (c) Memory, in Mb, needed by the fast multipole and precorrected-FFT algorithms. (d) Product of (b) and (c). Note the speed-memory product is fairly independent of grid size.

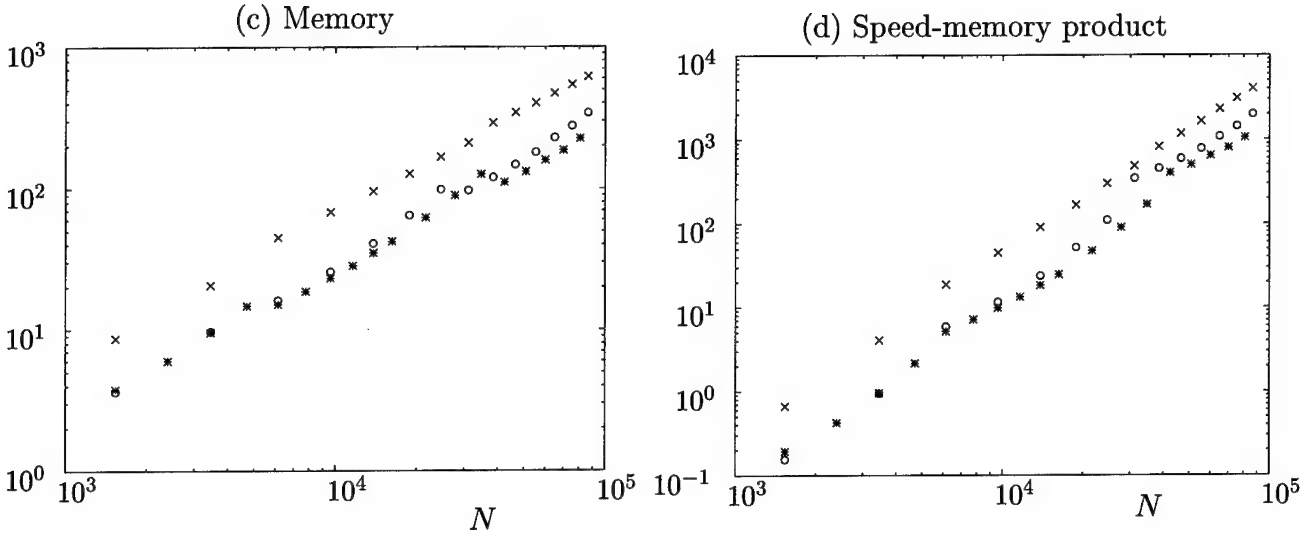
The fast multipole algorithms used in the FASTCAP program compute matrix-vector products in $O(n)$ operations regardless of the distribution of panels on the discretized surfaces [16], but this is not true of the precorrected-FFT method. As described, the use of the FFT implies that the algorithm computes matrix-vector products in at best $O(n \log n)$ operations, and attains this optimum only for fairly homogenous distributions of panels (see Chapter 4). That is, for problems where the panels are distributed in a roughly uniform manner throughout space, the precorrected-FFT method should be efficient. In contrast, for inhomogeneous problems which consist of clusters of panels separated by large areas of open space, inefficiency may be expected. Therefore it is important to quantify the performance penalty induced in the precorrected-FFT method by problem inhomogeneity.

A simple approach to generating an example which is inhomogenous is to refine the discretization of a cube. The cube example is intended to serve as a model for typical boundary-element discretizations of surfaces. As the discretization is refined, problems with increasing numbers of panels will be generated. The precorrected-FFT algorithm must place grid charges in the empty interior of the cube, which causes the CPU time and memory required by the algorithm to increase faster than $O(n \log n)$. As n increases, relatively more panels are near the edges of the cube relative to the interior, i.e., the problem inhomogeneity increases. Thus, at some large n, the fast multipole methods will be superior to the precorrected-FFT method. We wish to determine how effective the precorrected-FFT method is for reasonable size problems, and at what n it would become advantageous to use the fast-multipole methods.

Figure 5-2 shows the comparison of the precorrected-FFT method at $p = 3$ to the fast-multipole based code FASTCAP, at $l = 2$, for the cube example. The discretization of the cube is refined to generate more panels, and the performance of the two codes compared as the problem size increases. Three figures are shown. Figure 5-2(b) shows the time required for each code to compute a matrix-vector product, Figure 5-2(c) shows the amount of memory needed by each code, and Figure 5-2(d) shows a figure of merit which is the product of required memory and the time needed for a potential calculation. The product is important to consider when analyzing the precorrected-FFT method because, as is clear from the figure, speed can be traded for memory by manipulating the size of the region the grid-charge approximation covers. The CPU and memory figures for the precorrected-FFT method are observed to grow irregularly with problem size. This is because our specific implementation of the method requires the number of grid-charges along one side of the computational domain to be a power of two. The solid line in the figures shows results when the number of grid charges along a side was selected to optimize (see Section 3.7) the speed-memory product, which is observed to grow smoothly. Two cases in Figure 5-2(a) are evident where the code would have been considerably faster had a different number of grid-charges been used. However, as Figure 5-2(b) shows, the memory required would have been greater in each case.

Analysis of the trend of Figure 5-2(a) reveals that the CPU time needed to solve the cube


(b) CPU time


(c) Memory

(d) Speed-memory product

FIGURE 5-3: The nonuniformly-discretized cube example. 86,400 panels were present in the discretization shown. (b)-(d): Computational resources needed for the fast multipole (x) and precorrected-FFT algorithms (o) to compute a matrix-vector product for the nonuniformly discretized cube geometry. (*) shows time needed by the precorrected-FFT algorithm for the uniform-cube discretization. (b) CPU time, in seconds, (c) Memory, in Mb, needed by the fast multipole and precorrected-FFT algorithms. (d) Product of (b) and (c).

problem grows as about $O(n^{1.15})$, where n is the number of panels, faster than the $O(n)$ expected asymptotically for the fast multipole method. However, for all the problems analyzed, the precorrected-FFT method was superior in terms of CPU time and memory required. We may obtain the approximate point at which the algorithms cross over by extrapolating the data in Fig. 5-2(c). Assuming that the CPU time and memory of the multipole method grow as $O(n)$, and that the CPU time and memory required by the precorrected-FFT method grows as $O(n^{1.2})$[38], then in terms of the speed-memory product the precorrected-FFT method will be superior to the fast-multipole method until n is, at least, several million panels. We estimate over 30 gigabytes of memory would be needed to solve such a problem.

Another type of inhomogeneity is commonly encountered in practical problems. In the first cube example, with the exception of a single smaller edge panel, the discretization was mostly uniform, i.e. all the panels were roughly the same size. For surfaces with edges and corners, however, the charge density varies more rapidly near the edges, and so it is often desirable to discretize the surface such that there are small panels near the edges and larger ones in the interior. To examine the effect of this type of discretization on the precorrected-FFT method, "cosine spacing" was used to achieve a non-uniform discretization. By this it is meant that a non-uniform distribution of $N+1$ nodes x_k on the interval $[0,1]$ was generated from

$$x_k = \cos\frac{k\pi}{2N} \qquad k = 0\ldots N \tag{5.4}$$

and the panel sizes derived from the resultant spacings $x_k - x_{k-1}$. As can be seen from Figure 5-3 (a), this generates a discretization with smaller panels near the edges of the cube, as needed to resolve the rapidly varying charge density. In Figure 5-3 (b)–(d), comparisons are made between the precorrected-FFT and fast multipole algorithms applied to the nonuniform cube discretization, as well as to the resources needed for the uniform cube. Consider in particular plot (d) which compares the speed-memory products. As the number of panels is increased, the non-uniform discretization becomes less homogeneous relative to the uniform one. Comparing the speed-memory products of the precorrected-FFT algorithm for the two different families of discretizations, we see that the precorrected-FFT algorithm is indeed less efficient on the non-uniformly discretized problems. Both CPU time and memory are observed to grow somewhat more rapidly with problem size for the non-uniform discretization compared to the uniform one. However, for the largest problems considered the increase in resources used is only about 20–30%. In fact even for the worst cases considered the precorrected-FFT method is still superior, in terms of the speed-memory product, to the fast multipole method. This is primarily due to its superior memory utilization.

The cube examples demonstrate that problems exist for which the precorrected-FFT algorithm is inferior to the fast-multipole methods. This example, however, is somewhat artificial, as very large capacitance extraction problems are not usually due to very fine discretizations of a few surfaces, but rather by fixing a discretization level, and solving problems which involve

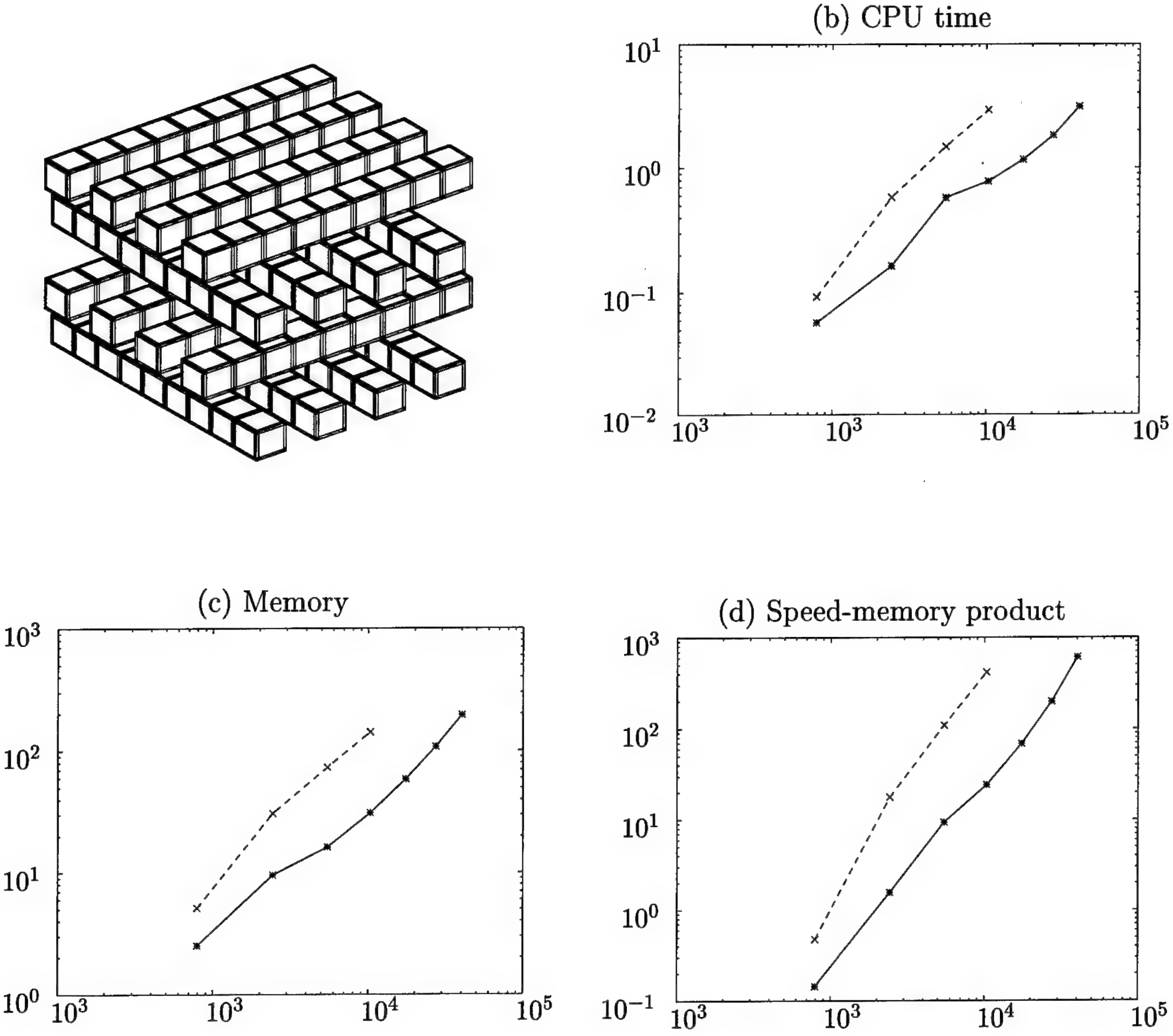


FIGURE 5-4: The bus crossing example. (a) Larger problems are generated by adding more bus lines. (b) CPU time, in seconds, needed for the fast multipole (dashed line connecting 'x') and precorrected-FFT algorithms (solid line connecting '*') to compute a matrix-vector product. (c) Memory, in Mb, needed by the fast multipole and precorrected-FFT algorithms. (d) Product of (b) and (c).

increasingly more complicated structures. For a situation which better models problems from VLSI interconnect analysis, consider a bus crossing example, as in Figure 5-4. In this example, a series of stacked bus problems are solved. The faces of each bus line are broken into quadrilateral sections, and the quadrilaterals discretized by division into a central panel and five edge panels. In order to generate larger and larger problems, we consider b levels of bus wires, each level having b wires.

From Fig. 5-4 it is clear that the computational cost of the algorithm grows nearly linearly with problem size, as predicted in Chapter 4. For the size problems considered, the precorrected-FFT method with $p = 3$ enjoys an advantage of more than a factor of three in terms of computational cost and roughly a factor of four in memory utilization over the fast multipole method using order-2 expansions. With these parameter values, however, from Figure 5-1 we expect the precorrected-FFT method to be considerably more accurate.

5.3 Realistic Examples

In this section we present results comparing the FASTCAP program to the precorrected-FFT method for computing capacitances of several three-dimensional geometries. As a preconditioner has not yet been implemented in the precorrected-FFT algorithm, all comparisons were performed without FASTCAP's preconditioner. Figure 5.3 shows four realistic three-dimensional structures: a woven bus structure, a bus crossing structure, a via structure, and part of an SRAM memory cell. We have compared the multipole-based code FASTCAP, using multipole expansion order $l = 2$, to the grid based methods with $p = 2$ and $p = 3$. To estimate the accuracy of the computed capacitances, we have compared the results to the grid-code run using $p = 6$, which we expect to introduce errors into the calculation which are very small compared to the $p = 2$, $p = 3$ grid codes or the multipole $l = 2$ code. As a check on this assumption we also performed the calculations using the fast multipole algorithm and sixth order multipole expansions. Taking the $p = 6$ capacitances to be exact, we have calculated both the maximum relative errors in the computed capacitance coefficients, as well as the maximum over all rows of the capacitance matrix of the largest error in the row as a fraction of that row's diagonal capacitance. Table 5-1 shows the computation times, memory required, and error estimates for each problem. All experiments were run on a DEC AXP3000/900, with 256 megabytes of physical memory.

The table indicates that multipole expansions of order 2 are usually enough to give relative accuracy of one percent or so in the calculated capacitances. In terms of relative errors in the computed capacitances, the $p = 2$ grid code appears to be comparable to the $l = 2$ multipole code, and somewhat inferior when the error is measured as a percentage of the diagonal capacitance. The $p = 3$ grid code clearly has uniformly superior error properties. These results are in accord with the sphere example considered previously in Figure 5-1. Note also Table 5-3 which compares the errors in final computed capacitances induced by the grid-projection scheme when the grid-collocation technique is used to generate interpolation operators to the errors induced by polynomial interpolation. Though the errors in these examples are small enough to make firm statements difficult, it seems that the polynomials are less accurate, especially for the woven bus example which generates the largest errors. For this example, it is interesting that both grid-projection schemes gave more accurate answers than the fast-multipole based codes. It is possible that this is because, even if all three schemes generated similar worst-case errors, the average errors for the grid-based algorithm will be lower because almost all the panels in the system are well-separated from the approximate grid charges. In contrast, in the fast multipole scheme, because of the multilevel hierarchy, for any given approximation of a panel charge, any panel is "close" to the approximating multipole expansion.

Table 5-2 shows explicit performance comparisons of the $l = 2$ multipole code to the $p = 2$ grid code, which has comparable accuracy, as well as to the more-accurate $p = 3$ grid codes. At $p = 3$, the precorrected-FFT method can be as much as four times faster and can use as little as one fifth the memory of FASTCAP. In terms of the speed-memory product, the grid-based code at $p = 3$ was superior by a factor ranging from four to twenty. At $p = 2$, the performance advantage of the grid-code was even more significant. The CPU advantage of the method ranged from nearly four to more than eight, the memory advantage from four to six, and the product from twelve to fifty-two.

The two final entries in Table 5-1 are worthy of note. Using the $p = 2$ grid representation, from which we expect about 2-4% accuracy, it was possible to analyze two very large problems. The first is a fifteen-by-fifteen wire woven bus crossing, shown in Fig. 5-6, which has over 80,000 panels in the discretization. The second is the cube, discretized into about 125,000 panels. The precorrected-FFT method was able to perform a single solution (one row in the capacitance matrix) in only about three minutes. More importantly, both problems could be solved in the available physical memory, which was not possible using FASTCAP.

We note that all the examples considered here generate fairly well-conditioned linear systems, so no preconditioner was necessary to secure convergence in a reasonable time. However, use of a preconditioner could further reduce the computation times required for the linear system solution, and enable solution of less well-conditioned problems.

Finally, we wish to emphasize that, regardless of whether the precorrected-FFT or fast-multipole based approaches are used, the advantage of the accelerated schemes over traditional algorithms is tremendous. Table 5-4 compares the computation time and memory needed by algorithms based on LU-factorization via Gaussian elimination, iterative solution using direct (explicit) matrix-vector products, and iterative solution using accelerated matrix-vector products, as described in this paper. The statistics for the direct algorithms were estimated by

---

[1] The large relative errors for the woven bus problems occur because some coupling capacitances are very small, 0.4% of the self capacitance for the 15x15 woven bus where the largest relative error occurs.

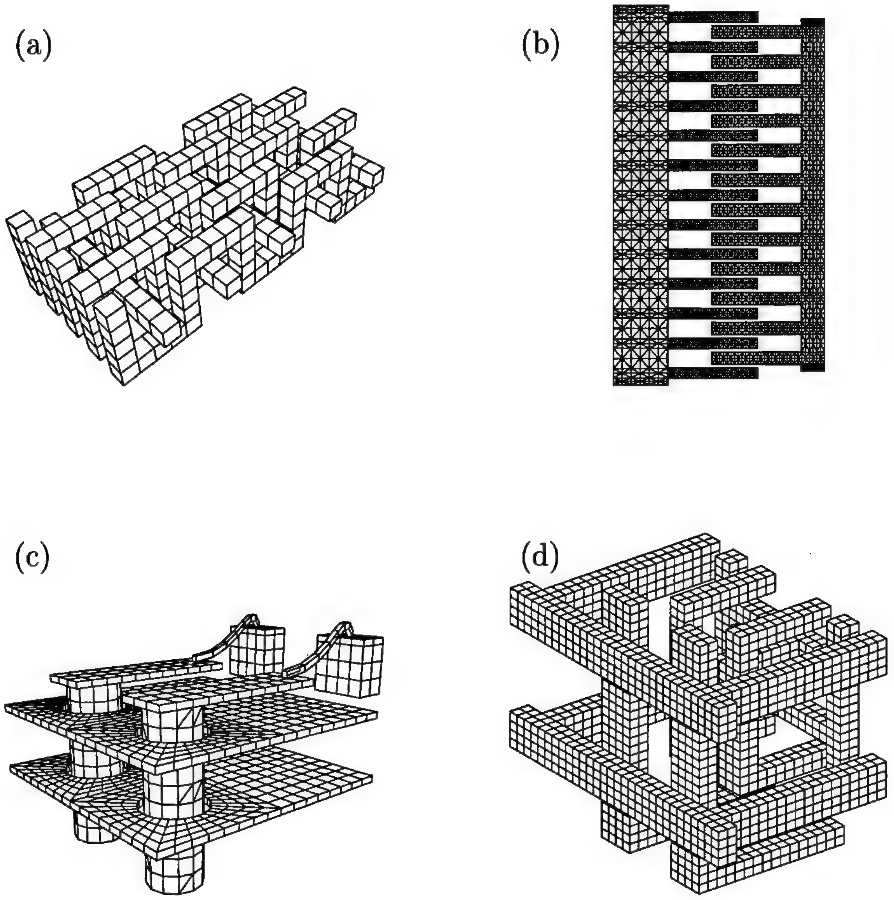


FIGURE 5-5: Several realistic capacitance extraction problems. (a) The woven bus example (woven5x5). (b) the comb drive example (comb). (c) The via example (via). (d) the SRAM example(SRAM).

Example[m]	Panels	Code	Setup	Solve	CPU	Memory	% err/rel	% err/diag
via [4]	6120	FFT[$p = 2$]	10.85	22.18	33.03	15.83	1.41	0.81
		FFT[$p = 3$]	10.60	61.96	72.56	20.92	0.068	0.026
		FASTCAP	18.56	101.32	119.88	55.82	0.369	0.152
woven5x5[10]	9360	FFT[$p = 2$]	5.8	125.2	131.0	20.35	7.06	1.65
		FFT[$p = 3$]	32.63	282.4	315.03	49.49	1.60	0.048
		FASTCAP	37.37	656.3	693.67	103.8	16.9	0.431
cube[1]	14406	FFT[$p = 2$]	4.80	8.80	13.6	25.03	0.105	0.105
		FFT[$p = 3$]	14.37	9.93	24.3	36.55	0.003	0.003
		FASTCAP	34.59	28.78	63.37	115.81	0.024	0.024
bus3x6[6]	6480	FFT[$p = 2$]	6.38	18.52	24.9	11.92	1.30	0.99
		FFT[$p = 3$]	6.59	44.70	51.29	15.74	0.033	0.015
		FASTCAP	21.75	184.2	205.95	75.49	1.89	0.164
bus3x8[6]	11520	FFT[$p = 2$]	5.67	52.1	57.77	20.22	1.10	0.416
		FFT[$p = 3$]	15.2	82.86	98.06	32.64	0.021	0.021
		FASTCAP	30.52	328.38	358.9	119.9	1.96	0.177
SRAM[6]	3944	FFT[$p = 2$]	3.92	11.83	15.75	7.71	0.90	0.45
		FFT[$p = 3$]	8.30	28.02	36.32	16.70	0.046	0.023
		FASTCAP	11.89	81.22	93.11	38.92	0.62	0.082
comb[3]	19424	FFT[$p = 2$]	28.9	91.5	120.3	50.3	2.11	1.74
		FFT[$p = 3$]	23.2	315.8	339.0	71.1	0.056	0.043
		FASTCAP	70.2	399.3	469.5	211.0	0.12	0.11
woven15[30]	82080	FFT[$p = 2$]	134.2	6152.8	6287.0	246.3	63.8[1]	1.81
cube[1]	126150	FFT[$p = 2$]	73.6	127.5	201.1	224.7	–	–

Table 5-1: Statistics for FASTCAP Order-2, Grid-2,3 codes for $1/r$ Green function. Setup, solve, and CPU times are in seconds on DEC AXP 3000/900, memory in megabytes. m is number of conductors in problem, each conductor requires a separate linear system solution.

Example	Grid Order	Setup	Solve	CPU	Memory	Product
via	$p = 2$	0.58	0.22	0.28	0.28	0.078
	$p = 3$	0.57	0.61	0.61	0.37	0.23
woven5x5	$p = 2$	0.16	0.19	0.19	0.20	0.037
	$p = 3$	0.87	0.43	0.45	0.48	0.22
cube	$p = 2$	0.14	0.31	0.21	0.22	0.046
	$p = 3$	0.42	0.34	0.38	0.32	0.12
bus3x6	$p = 2$	0.29	0.10	0.12	0.16	0.019
	$p = 3$	0.30	0.24	0.25	0.21	0.052
bus3x8	$p = 2$	0.19	0.16	0.16	0.17	0.027
	$p = 3$	0.50	0.25	0.27	0.27	0.074
SRAM	$p = 2$	0.33	0.15	0.17	0.20	0.034
	$p = 3$	0.70	0.34	0.39	0.43	0.17
comb	$p = 2$	0.41	0.23	0.26	0.24	0.062
	$p = 3$	0.33	0.80	0.72	0.34	0.24

Table 5-2: Comparison of FASTCAP and grid codes. Figures are ratios of required resources. "Product" is product of CPU and memory figure.

Example	Colloc % err rel	Poly % err rel
via	0.068	0.090
woven5x5	1.6	5.0
cube	0.003	.0001
bus3x6	0.033	0.049
bus3x8	0.021	0.066
SRAM	0.046	0.2
comb	0.056	0.072

Table 5-3: Comparison of errors in capacitances induced by using grid-collocation operators for interpolation and polynomials for interpolation. Both columns use grid-collocation for the grid projection step.

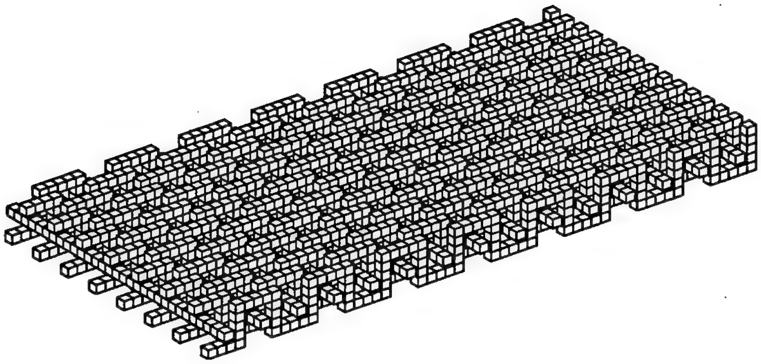


FIGURE 5-6: The large woven bus structure. The 30 conductor structure is formed from fifteen conductors woven around fifteen straight conductors. The actual discretization is finer than shown in the above figure. There are 82,080 panels in the actual discretization.

Example		CPU Usage				Memory Usage	
Name	Panels[cond]	P/FFT	FASTCAP	Direct Iterative	LU Decomp	P/FFT	Direct
via	6120[4]	1.1 min	2.0 min	(5.6 min)	(1.9 hrs)	21 Mb	(286 Mb)
woven5x5	9360[10]	5.2 min	12 min	(42 min)	(6.9 hrs)	50 Mb	(668 Mb)
woven15	82080[30]	1.7 hrs	-	(11.5 days)	(194 days)	246 Mb	(50.2 Gb)
cube	126150[1]	3.3 min	-	(8.4hrs)	(2.7 yrs)	225 Mb	(119 Gb)

Table 5-4: Comparision of capacitance extraction algorithms. Figures in parentheses are estimates.

extrapolating timings of computations performed by MATLAB, and by assuming that storage is in double-precision floating point words. For the largest problems, the speedups can be two orders of magnitude over iterative solution with direct products, and nearly *six* orders of magnitude over Gaussian-elimination, with memory savings a factor of 500.

6

The Laplace Equation in Stratified Media

6.1 Motivation

Many applications requiring the solution of the Laplace equation have more complicated boundaries and/or boundary conditions than free space. It is sometimes useful to formulate the problem by calculating a Green function which accounts for the special geometry of a system, thereby removing part of the problem domain from consideration. For an arbitrary Green function $g(\vec{x}|\vec{x}')$ the precorrected-FFT algorithm is not applicable since the grid-potential calculation cannot be expressed as a discrete convolution. However, an important special case where the pre-corrected FFT algorithm can still be applied occurs in a system where the Green function is modified by the presence of planar structure, for example, ground planes or planar dielectric interfaces. To illustrate the problem and its solution, first consider the case of a set of conducting bodies over a single ground plane located at $z = 0$ and extending to infinity in the x and y directions.

By the method of images, the potential at $\vec{x} = (x, y, z)$ due to a single charge at $\vec{x}' = (x', y', z')$ is

$$\psi(x,y,z) = g(x-x', y-y', z-z') - g(x-x', y-y', z+z') \tag{6.1}$$

where g is the free-space Green function. The difficulty for the precorrected-FFT method is that the second term depends on $z + z'$, a general occurrence for problems with planar interfaces. The matrix mapping grid charges to grid potentials is the sum of a matrix with block-Toeplitz structure, corresponding to the first term of Eq. 6.1, and a matrix with block-Hankel structure, corresponding to the second term of Eq. 6.1. The Toeplitz-like part of the matrix corresponds to the discrete convolution with the free-space Green function, and can be treated directly with the FFT as described above. Because a Hankel matrix is related to a Toeplitz matrix via a permutation matrix[36] which is simple to compute, multiplication by a Hankel matrix may also be done in $O(N \log N)$ time via the FFT. Furthermore, as we will see, the permutation matrix may be represented in Fourier space so that multiplication of a vector by the sum of a Hankel and Toeplitz matrix can be performed using a single forward and inverse FFT pair. Thus, at each iteration of GMRES, the only additional computation required to incorporate a modified Green function is multiplication in Fourier space by a diagonal matrix and a permutation matrix, which requires negligible additional computation time.

An important special case of the general class of Laplace problems is the case of piecewise-constant dielectrics. In such a system, the potential ψ generated by a charge density $\rho(\vec{x})$ in the material with dielectric constant ϵ_j satisfies, for $\vec{x}$ not on an interface,

$$\nabla^2 \psi(\vec{x}) = -\frac{\rho(\vec{x})}{\epsilon_j} \tag{6.2}$$

where ϵ_j is the dielectric constant in the jth dielectric. At a dielectric interface, the potential ψ satisfies the boundary conditions[33]

$$\epsilon_+ \frac{\partial \psi}{\partial n^+} = \epsilon_- \frac{\partial \psi}{\partial n^-} \tag{6.3}$$

$$\psi(+) = \psi(-) \tag{6.4}$$

where $\partial\psi/\partial n^+, \partial\psi/\partial n^-$ denote the normal derivatives of the potential ψ taken from the side of the interface with dielectric constant ϵ^+, ϵ^- respectively, and $\psi(+)$ and $\psi(-)$ the potential on opposite sides of the interface.

Two different multipole accelerated algorithms have been proposed for problems with dielectric interfaces. In [56], based on [9], the interface is replaced by a fictitious charge distribution constructed to satisfy the interface boundary conditions (6.3, 6.4). The introduction of the fictitious charges requires the discretization of the dielectric interface. There are four potential problems with discretizing the interface. First, the accurate discretization of the interfaces may itself be problematic. It is possible to construct adaptive discretization procedures[57], but such algorithms generally have a computational cost several times that requires for a single integral equation solution. Such algorithms are, moreover, not widely available. Second, discretization of the interfaces requires increasing the number of panels in the discretized integral-equation solver. For problems with small numbers of widely-spaced conductors, the number of additional panels required to discretize the interface could significantly exceed the number of panels on the conductors of interest, resulting in an unacceptably high computational cost. Third, as we will see later, the addition of discretized interfaces can worsen the conditioning of the matrix equation that must be solved to obtain the surface charge densities. This results in increased computational cost and possible decreased accuracy of the solution. Finally, straightforward integral equation formulations for problems with large ratios of dielectric constant can introduce scaling difficulties that make accurate determination of the charge on the original conductors difficult[58, 59]. On the other hand, when solving problems involving dielectric interfaces that are not planar, the appropriate Green functions are not in general easily computable, and so the equivalent-charge formulations are necessary, as they have the advantage that the dielectic interfaces can have any shape. In that case, once the fictitious charge has been introduced, the problem is reduced to a free-space Laplace boundary-value problem, to which the precorrected-FFT method can be applied with advantages and drawbacks as discussed in Chapter 5.

In reference [15], the discretization of dielectric interfaces is avoided by working directly with the Green function associated with the potential of a charge in a system of layered dielectrics. The Green function is approximated by a weighted collection of image charges. These fictitious image charges can be incorporated in a modified version of the fast multipole algorithm in order to calculate the potential of the collection of panel charges. This approach avoids the disadvantages associated with discretizing the dielectric interfaces. However, the multipole domain must be extended to encompass the regions occupied by the fictitious image charges. The computational cost, in time and memory, is thus proportional to the number of panels times the number of images. The number of images may be quite large (twenty or thirty). The authors of [15] restricted their study to cases with two interfaces where the conductors lie strictly in one layer. The algorithm can in principle be extended to accomodate more interfaces, but such an algorithm would likely be very costly and unwieldy.

In this chapter, we describe how the precorrected-FFT algorithm can be used to evaluate the potential of a system of charges in a system of piecewise-constant dielectrics, with infinite planar interfaces. Unlike in [15], the cost of the potential evaluation is shown to be essentially independent of the Green function used. The distribution of conductors among multiple dielectric layers can introduce inefficiencies relative to free-space problems, however. We restrict our study to problems that have at most two planar interfaces, though in principle the methods discusses herein can be extended to an arbitrary number of interfaces. For simplicity we will limit our exposition to cases with conductors lying only in the top two spaces; the extension to conductors in all three spaces is trivial.

6.2 Calculating Green functions for planar geometries

6.2.1 Green function derivation

Consider representing the potential by the integral

$$\psi(\vec{x}) = \int \sigma(\vec{x}')g(\vec{x},\vec{x}')dS \tag{6.5}$$

If the $g(\vec{x},\vec{x}')$ satisfies

$$\nabla^2 g(\vec{x},\vec{x}') = -\frac{\delta(\vec{x},\vec{x}')}{\epsilon} \tag{6.6}$$

then ψ will satisfy (6.2). To simplify notation, for the time being consider the problem of a point charge at the origin,

$$\nabla^2 g(\vec{x}) = -\frac{\delta(\vec{x})}{\epsilon}. \tag{6.7}$$

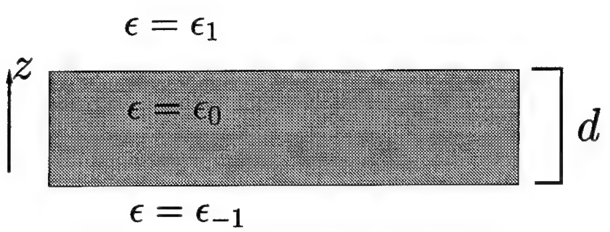


FIGURE 6-1: Two-interface layered dielectric problem. Interfaces extend to infinity in x and y directions.

It will be convenient to introduce the Fourier transform $\tilde{g}$ of g,

$$g(\vec{x}) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} d\alpha d\beta \; e^{-j\alpha x} e^{-j\beta y} \tilde{g}(\alpha, \beta, z) \tag{6.8}$$

In a material with dielectric constant ϵ, $\tilde{g}$ satisfies the equation

$$\left(\frac{\partial^2 \tilde{g}(\alpha, \beta, z)}{\partial z^2} - \alpha^2 - \beta^2 \right) = -\frac{\delta(z)}{\epsilon} \tag{6.9}$$

and the boundary conditions (6.4, 6.3). Letting $\gamma^2 = \alpha^2 + \beta^2$, the differential equation for $\tilde{g}$ can be written

$$\left(\frac{\partial^2}{\partial z^2} - \gamma^2 \right) \tilde{g}(\gamma, z) = -\frac{\delta(z)}{\epsilon}. \tag{6.10}$$

For a single uniform dielectric, $\tilde{g}$ can be determined by solving

$$\left(\frac{\partial^2}{\partial z^2} - \gamma^2 \right) \tilde{g}(\gamma, z) = 0 \tag{6.11}$$

with the boundary conditions

$$\tilde{g}(\gamma, 0^+) = \tilde{g}(\gamma, 0^-) \tag{6.12}$$

$$\frac{\partial \tilde{g}(\gamma, z)}{\partial z^+} - \frac{\partial \tilde{g}(\gamma, z)}{\partial z^-} = -\frac{1}{\epsilon} \tag{6.13}$$

This has the solution

$$\tilde{g}(\gamma, z) = \frac{1}{2\gamma\epsilon} e^{\gamma |z|} \tag{6.14}$$

Since we know that, for a infinite uniform dielectric,

$$g(\vec{x}, \vec{x}') = \frac{1}{4\pi\epsilon ||\vec{x} - \vec{x}'||} \tag{6.15}$$

we can conclude, by uniqueness of solutions to the Laplace equation, that the inverse Fourier transform of $\tilde{g}$ is

$$\frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} d\alpha d\beta e^{-j\alpha(x-x')} e^{-j\beta(y-y')} \frac{1}{2\gamma\epsilon} e^{\gamma |z-z'|} = \frac{1}{4\pi\epsilon ||\vec{x} - \vec{x}'||} \tag{6.16}$$

Consider the three-layer dielectric structure shown in Figure 6.2.1. For a source located at $z', 0 \leq z' \leq d$, (in the middle layer), the Fourier-domain Green function is

$$\tilde{g}(\gamma,z,z') = \frac{e^{-\gamma(z-z')}}{2\gamma\epsilon_0} + \frac{S(\gamma,d)}{2\gamma\epsilon_0}\left[R_+R_-e^{-\gamma(z-z'+2d)} + R_-e^{-\gamma(z+z')} + R_+R_-e^{-\gamma(z+z')} + R_+e^{-\gamma(z-z')}\right] \tag{6.17}$$

for $z > d$, and for $0 \leq z \leq d$,

$$\tilde{g}(\gamma,z,z') = \frac{e^{-\gamma(z-z')}}{2\gamma\epsilon_0} + \frac{S(\gamma,d)}{2\gamma\epsilon_0}\left[R_+R_-e^{-\gamma(2d-z+z')} + R_-e^{-\gamma(z+z')} + R_+R_-e^{-\gamma(2d-z'+z)} + R_+e^{-\gamma(2d-z-z')}\right] \tag{6.18}$$

where

$$R_+ = \frac{\epsilon_0 - \epsilon_1}{\epsilon_0 + \epsilon_1} \tag{6.19}$$

$$R_- = \frac{\epsilon_0 - \epsilon_{-1}}{\epsilon_0 + \epsilon_{-1}} \tag{6.20}$$

and

$$S(\gamma,d) = \frac{1}{1 - R_+R_-e^{-2\gamma d}} \tag{6.21}$$

For a source at $z' > d$, we have for $z > d$

$$\tilde{g}(\gamma,z,z') = \frac{1}{2\gamma\epsilon_1}\left[e^{-\gamma|z-z'|} - e^{\gamma(z+z'-2d)} + T_+R_-S(\gamma,d)e^{-\gamma(z+z')} + T_+S(\gamma,d)e^{-\gamma(z+z'-2d)}\right] \tag{6.22}$$

$$\tilde{g}(\gamma,z,z') = \frac{T_+}{2\gamma\epsilon_1}\left[e^{-\gamma(z'-z)} + R_-S(\gamma,d)e^{-\gamma(z'+z)} + S(\gamma,d)R_+R_-e^{-\gamma(2d-z+z')}\right] \qquad 0 \leq z \leq d \tag{6.23}$$

where

$$T_+ = \frac{2\epsilon_1}{\epsilon_0 + \epsilon_1} \tag{6.24}$$

Note that this formulation can account for the presence of a ground plane at one of the interfaces by taking $\epsilon_1 \to \infty$ or $\epsilon_{-1} \to -\infty$, as appropriate, in which case $R_+ = -1$ or $R_- = -1$ respectively. Care must be taken if both $R_+, R_- \to -1$ as then the function $S(\gamma, d)$ is singular at $\gamma = 0$. A symmetry plane can also be accomodated by taking $\epsilon_1 \to 0$ or $\epsilon_{-1} \to 0$ as needed.

The key to the applicability of the precorrected-FFT method is that, as is clear from inspection of Equations 6.17-6.23 and the linearity of the Fourier transform, each of the Green functions can be decomposed into two terms,

$$g(\vec{x},\vec{x}') = f(x-x', y-y', z-z') + h(x-x', y-y', z+z') \tag{6.25}$$

6.2.2 Approximating the Green functions

If it were not for the factor $S(\gamma, d)$, all the terms in the Green functions above would be exponentials. As the inverse Fourier transform of $e^{-\gamma|z|}$ is the potential due to a point-charge, if the function $S(\gamma, d)$ could be represented as a sum of exponentials, then the Green function could be represented as a sum of the potentials of weighted point-charges. In fact the inverse Fourier transform of each of the functions corresponds to the potential of a infinite series of images. In general, for a problem with n interfaces, the Green function can be expressed in terms of an $(n-1)$-fold nested infinite series of image charges. These series may converge very slowly, however, and even if they converge rapidly, if the number of interfaces is large, summing the contribution from all the significant images may be expensive. Therefore it is worthwhile to look for approximations to the Green functions by a finite number of images.

$S(\gamma, d)$ can be written in terms of the function

$$F(u) = \frac{Ke^{-u}}{1 - Ke^{-u}} \tag{6.26}$$

by

$$S(\gamma, d) = 1 + F(\gamma d) \tag{6.27}$$

where $K = R_+ R_-$. For the two-interface problem, finding an image approximation to the Green functions is then equivalent to approximating $F(u)$ by

$$F(u) \simeq \sum_{i=1}^{N} a_i e^{b_i u} \tag{6.28}$$

and generally we will want to require that the b_i be real and negative[60]. It is of course also desirable that N be small. We solve the approximation problem by collocating at judiciously chosen points $u_j, j = 1 \ldots 2N$. That is, we require

$$F(u_j) = \sum_{i=1}^{N} a_i e^{b_i u_j} \tag{6.29}$$

for each j. The resulting system of nonlinear equations is solved via a continuation technique[61], with K as the continuation variable. Many other methods to fit a sum of exponentials to a function have been proposed, such as the classical Prony's method[62].

6.3 Algorithmic Implications

6.3.1 The interpolation/projection operators

We note that the Green function for the layered media case may be obtained by the method of images, and thereby decomposed into several terms, each interpretable as potential due to a charge in free-space. The precise analytic form of the decomposition, such as into multiply-infinite series, may be cumbersome, but what is important is to note that such a decomposition exists. Since the potential is due to a set of image charges each having the free-space Green function, Theorem 2.1 implies that the projection of charge onto the grid and interpolation of grid potentials can be done by using the *same* operators as for charges in free space. Further, we note that the distance to any image charge will be greater than the distance to the actual charge. Thus the error bounds derived in Chapter 2 for representation of Laplace potentials in free space hold for the grid-collocation scheme in layered media as well. Therefore, to incorporate layered media into a capacitance extraction code, the only change in the precorrected-FFT method is in computation of grid potentials due to grid charges.

A more insightful viewpoint can be obtained by noting that one interpretation of the grid-collocation approach of Chapter 2 is that it is a means to obtain interpolation operators which are highly accurate for harmonic functions. The precise form of source of the harmonic potential is not particularly important. Since the Green functions satisfy the Laplace equation except at sources and layer boundaries, neither of which are treated by the grid, any operator which accurately interpolates such functions can be used for projection and interpolation in the grid-base scheme.

6.3.2 Modifications to the FFT-based convolution

Two modifications to the convolution step in precorrected-FFT approach are needed when moving from free-space problems to problems with planar interfaces. First, the code must be modified to account for the presence of Hankel-like structure. To see how this can be done, consider a one dimensional example. Define the matrices $T, H \in \mathbf{R}^{n \times n}$ by

$$T_{ij} = i - j \qquad H_{ij} = i + j - 5 \tag{6.30}$$

The $n = 4$ matrices are

$$T = \begin{bmatrix} 0 & -1 & -2 & -3 \\ 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \qquad H = \begin{bmatrix} -3 & -2 & -1 & 0 \\ -2 & -1 & 0 & 1 \\ -1 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix} \tag{6.31}$$

For a vector $x \in \mathbf{R}^n$, to compute Tx we first embed T into a circulant matrix $C_T \in \mathbf{R}^{2n \times 2n}$,

$$C_T = \left[\begin{array}{cccc|cccc} 0 & -1 & -2 & -3 & z & 3 & 2 & 1 \\ 1 & 0 & -1 & -2 & -3 & z & 3 & 2 \\ 2 & 1 & 0 & -1 & -2 & -3 & z & 3 \\ 3 & 2 & 1 & 0 & -1 & -2 & -3 & z \\ \hline z & 3 & 2 & 1 & 0 & -1 & -2 & -3 \\ -3 & z & 3 & 2 & 1 & 0 & -1 & -2 \\ -2 & -3 & z & 3 & 2 & 1 & 0 & -1 \\ -1 & -2 & -3 & z & 3 & 2 & 1 & 0 \end{array} \right] \tag{6.32}$$

where z denotes an arbitrary entry. As C_T is circulant, it has the eigendecomposition $C_T = F^H \Lambda F$, where F is the discrete Fourier transform matrix normalized by $\sqrt{n}$, i.e. $F^H F = I$. The diagonal entries of Λ are Fc, where c is the first column of C_T. To compute Tx using C_T, x is first embedded into a vector x_P of length $2n$,

$$x_P = \begin{bmatrix} x \\ 0 \end{bmatrix} \tag{6.33}$$

a step sometimes referred to as "zero-padding." Now we note that

$$\begin{bmatrix} Tx \\ y \end{bmatrix} = \left[\begin{array}{cccc|cccc} 0 & -1 & -2 & -3 & z & 3 & 2 & 1 \\ 1 & 0 & -1 & -2 & -3 & z & 3 & 2 \\ 2 & 1 & 0 & -1 & -2 & -3 & z & 3 \\ 3 & 2 & 1 & 0 & -1 & -2 & -3 & z \\ \hline z & 3 & 2 & 1 & 0 & -1 & -2 & -3 \\ -3 & z & 3 & 2 & 1 & 0 & -1 & -2 \\ -2 & -3 & z & 3 & 2 & 1 & 0 & -1 \\ -1 & -2 & -3 & z & 3 & 2 & 1 & 0 \end{array}\right] \begin{bmatrix} x \\ 0 \end{bmatrix} = C_T x_P = F^H \Lambda F x_P \tag{6.34}$$

where $y \in \mathbf{R}^n$ is a vector to be discarded. To perform this computation, Fx_P and Fc are computed using the FFT. The multiplication by F^H is simply the inverse-FFT.

Let x' denote x with the entries written *backwards*, i.e.

$$x'_k = x_{n-k+1} \tag{6.35}$$

Likewise, let H' denote H with the columns similarly reordered such that $H'x' = Hx$ for any x. We have

$$H' = \begin{bmatrix} 0 & -1 & -2 & -3 \\ 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 \\ 3 & 2 & 1 & 0 \end{bmatrix} = T \tag{6.36}$$

Thus H is related to T by a permutation matrix P,

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.37}$$

and in fact any Hankel matrix is related to a Toeplitz matrix by a permutation matrix P which has ones along the principal anti-diagonal. Note that $x' = Px$. Thus, $Hx = H'x' = HPPx = TPx$. So we see that multiplication by either a Toeplitz or a Hankel matrix can be performed by the FFT. What is less obvious is that multiplication by matrices of the form $T + H$, with T and H arbitrary Toeplitz and Hankel matrices respectively, can be performed via a single FFT. This is possible because it happens that the FFT of x' can be easily computed from the FFT of x.

Take $x \in \mathbf{R}^N$ and let $x' \in \mathbf{R}^N$ be the the reversed vector, i.e.

$$x'_n = x_{N-n-1} \tag{6.38}$$

Let X, X' be the discrete Fourier transforms of x, x' respectively:

$$X_k = Fx = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n/N} \tag{6.39}$$

$$X'_k = Fx' = \sum_{n=0}^{N-1} x'_n e^{-2\pi ikn/N} = \sum_{n=0}^{N-1} x_{N-n-1} e^{-2\pi ikn/N} \tag{6.40}$$

Letting $m = N - n - 1$,

$$X'_k = \sum_{m=0}^{N-1} x_m e^{-2\pi ik(N-m-1)/N} \tag{6.41}$$

or

$$(FPx)_k = X'_k = e^{2\pi ik/N} X^*_k \tag{6.42}$$

where X^*_k denotes complex conjugate of X_k. An additional factor is needed to take care of the zero-padding. If

$$x'_P = \begin{bmatrix} x' \\ 0 \end{bmatrix} \tag{6.43}$$

then the desired transform is

$$(Fx'_P)_k = (-1)^k e^{2\pi ik/N} (Fx_P)^*_k \tag{6.44}$$

Thus we see that the cost of computing the necessary auxiliary transform is quite small relative to the FFT cost, so the cost of multiplication by the sum of a Hankel and a Toeplitz matrix is only marginally greater than the cost of multiplication by a Toeplitz matrix.

The second issue arising in the layered media problem is that the Green function is now only piecewise-C^∞ away from the source point, whereas in free-space it was C^∞ everywhere away from the source point. Thus, if there are k layers which contain conductors, there are k^2 separate pieces to the Green function which need to be considered. If there are conductors in only one layer the situation is substantially the same as for free-space. Otherwise, k separate grids must be introduced, each lying strictly in one layer. Each grid must lie strictly in a single layer as, due to the discontinuity in the derivative of the potential at a dielectric interface, interpolation across interfaces will not be accurate. There are then k^2 separate Toeplitz and Hankel matrices which correspond to the pairwise grid-potential contributions from each layer to each other layer.

Let h_{mn}, t_{hm} denote the Hankel and Toeplitz parts, respectively, of the Green function for a source in layer n and observation point in layer m. The computation of the grid potential ψ_m for layer m becomes

$$\hat{\psi}_m(i,j,k) = \sum_n \sum_{i',j',k'} \left(t_{mn}(i-i', j-j', k-k') + h_{mn}(i-i', j-j', k+k')\right) \hat{q}_n(i',j',k') \tag{6.45}$$

Letting P denote the sparse operator which is the three-dimensional analogue of the Fourier-space reversal operation of Eq. 6.44, the convolution step of the algorithm becomes

```
/* Modified Convolution Step */
  for each layer $m$
    Compute $\hat{\mathbf{Q}}_{\mathbf{m}} = \mathrm{FFT}(\hat{q}_m)$
```

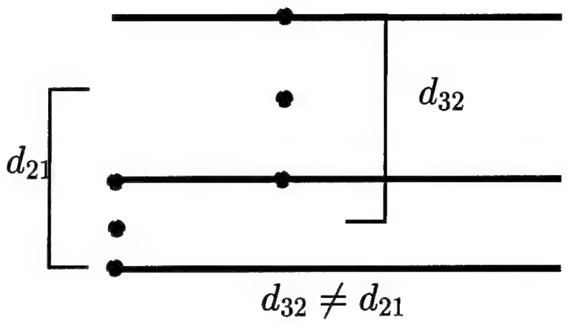


FIGURE 6-2: The grid spacing difficulty. If the grids in two layers have unequal spacing, $d_{32} \neq d_{21}$ and the grid-grid mapping matrices H_{mn} will not have Toeplitz structure.

```
Compute $\hat{\mathbf{\Psi}}_{\mathbf{n}} = \sum_m \tilde{\mathbf{H}}_{\mathbf{mn}}(I+P)\hat{\mathbf{Q}}_{\mathbf{n}}$
for each layer $m$
  Compute $\hat{\psi}_m = \mathrm{FFT}^{-1}(\hat{\mathbf{\Psi}}_{\mathbf{m}})$
```

6.3.3 Constraints on grid size and spacing.

In the free-space case the number grid points in each direction (n_x, n_y, n_z) and the spacing Δ of the grid charge points is purely a matter of computational efficiency (at least, for the Laplace type kernels this chapter is concerned with). This is essentially true in the layered media problem, as long as all the conducting surfaces lie in the same dielectric layer. Of course care must be taken to insure that interpolation across an interface is not attempted. When the conductors lie in different layers, the situation becomes more complicated. Figure 6.3.2 illustrates the principle difficulty. In order for the grid-potential matrices to have the appropriate Toeplitz structure, the grid spacing must be the same in each layer. In addition, in order to be able to exploit the FFT, the number of grid points (n_x, n_y, n_z) must be the same for the grids in each layer. This can result in computational inefficiency.

For each layer m let d_m denote the maximum difference in z-coordinate between any two points on conductors lying in layer m. For example, in Fig. 6.2.1, $d_0 \leq d$. We must have

$$n_z \Delta \geq \max_m d_m \tag{6.46}$$

for the grid to cover the conductors. If $d_i \ll \max_m d_m$ for some layer i, there will be many points in the grid of layer i which do not represent any charge density. Thus, we expect that the precorrected-FFT algorithm will be less efficient in layered media, for any given geometry, than its free-space counterpart.

Example[m]	Panels	Code	Setup	Solve	CPU	Memory
via [4]	6120	FFT[LM]	19.6	82.3	102.1	24.2
		FFT[FS]	10.60	61.96	72.56	20.92
		FASTCAP[FS]	18.56	101.32	119.88	55.82
woven5x5[10]	9360	FFT[LM]	78.9	368.7	447.6	61.1
		FFT[FS]	32.63	282.4	315.03	49.49
		FASTCAP[FS]	37.37	656.3	693.67	103.8
bus3x8[6]	11520	FFT[LM]	42.5	120.5	163.0	44.3
		FFT[FS]	15.2	82.86	98.06	32.64
		FASTCAP[FS]	30.52	328.38	358.9	119.9
SRAM[6]	3944	FFT[LM]	23.6	38.7	62.4	21.6
		FFT[FS]	8.30	28.02	36.32	16.70
		FASTCAP[FS]	11.89	81.22	93.11	38.92
comb[3]	19424	FFT[LM]	147.5	255.3	402.7	108.3
		FFT[FS]	23.2	315.8	339.0	71.1
		FASTCAP[FS]	70.2	399.3	469.5	211.0

Table 6-1: Statistics for FFTCAP-layered media code, FFTCAP free-space code, and FASTCAP. Setup, solve, and CPU times are in seconds on DEC AXP 3000/900, memory in megabytes. m is number of conductors in problem, each conductor requires a separate linear system solution. The LM code used the ground-plane Green function, the FS codes use the free-space Green function, and have no equivalent-charge discretized-interfaces for these problems.

6.4 Examples

First we demonstrate that the modified precorrected-FFT algorithm is an efficient approach for realistic engineering geometries. A single interface (a ground plane) was added to some of the example problems from Section 5.3. Table 6-1 shows the computational resources required by the free-space precorrected-FFT method [FS], the layered-media precorrected-FFT code [LM], and the fast-multipole based FASTCAP. From Table 6-1 it is clear that utilization of the layered media Green function does not result in a significant performance penalty, at least for the case when all the panels are in the same dielectric layer. The increases in resource usage are primarily due to inefficiencies introduced by the additional code needed to manipulate Hankel matrices and handle the various different possible dielectric configurations.

Next we compare the precorrected-FFT method with layered-media Green functions to the competing approach of discretizing the dielectric interface. For this study, we take as a canonical problem the calculation of the capacitances of two spheres. The spheres have radius 1m. Their centers are located 1.05m above the plane, and 3m from each other. The discretized interface version of this problem is shown in Figure 6-3.

To generate appropriate discretizations, we first consider the case of two spheres over a ground plane. First an adequate discretization of the sphere is generated by uniformly refining

Example	Code	Setup	Solve	Memory
via	FFT	1.85	1.33	1.16
	FMM	1.06	0.81	0.43
woven5x5	FFT	2.42	1.30	1.23
	FMM	2.1	0.56	0.59
bus3x8	FFT	2.8	1.45	1.36
	FMM	1.4	0.37	0.37
SRAM	FFT	2.84	1.38	1.29
	FMM	1.98	0.48	0.56
comb	FFT	6.6	0.81	1.52
	FMM	2.1	0.64	0.51

Table 6-2: Comparison of layered-media code to free-space FFTCAP and FASTCAP codes for the problems of Table 6-1. Figures are ratios of resources required by the layered media code to those needed by the free-space precorrected-FFT code (FFT in table) or the free-space FASTCAP (FMM) code.

the discretization until the self and mutual coupling capacitances of the spheres converges to within a tolerance of 1% of the diagonal value. Next the ground plane is added, and the discretization (again uniform) refined until 1% relative convergence is achieved. Finally, using the same spacing in the ground plane, the ground plane width is increased to convergence. The final discretization had 1600 panels in the ground plane and about 880 on each sphere.

Table 6-3 shows the relative computational resources required by the various approaches. While, for a given structure, the layered-media code requires more computational effort than the free-space code, the increased requirements are more than compensated by the extra work required when interfaces must be discretized. Due to the larger number of panels present when interfaces must be discretized, the time required for a matrix-vector product increases considerably. In addition, the matrix appears to be less well-conditioned as indicated by the increased number of iterations required for the GMRES solver to converge. Note that the time required by the fast multipole algorithm to compute a matrix-vector product with a *single* dielectric interface is almost six times more than the precorrected-FFT/layered media code requires. Even for the case of a symmetry plane, which is handled in the free-space codes by solving the capacitance problem for four spheres, the layered-media code gives a considerable savings in time and required memory.

Table 6-4 shows the computed capacitances of two spheres lying in a dielectric half-space. ϵ_+ is the dielectric constant in the upper half-space where the conducting hemispheres are located. ϵ_- is the dielectric constant in the other half space. Even for very large values of the dielectic constant, the computed capacitances seem to be well-behaved. Table 6-4 also shows the number of matrix-vector vector products needed for the two linear system solutions. As before, the free-space code with discretized interfaces needs more iterations to converge. Note

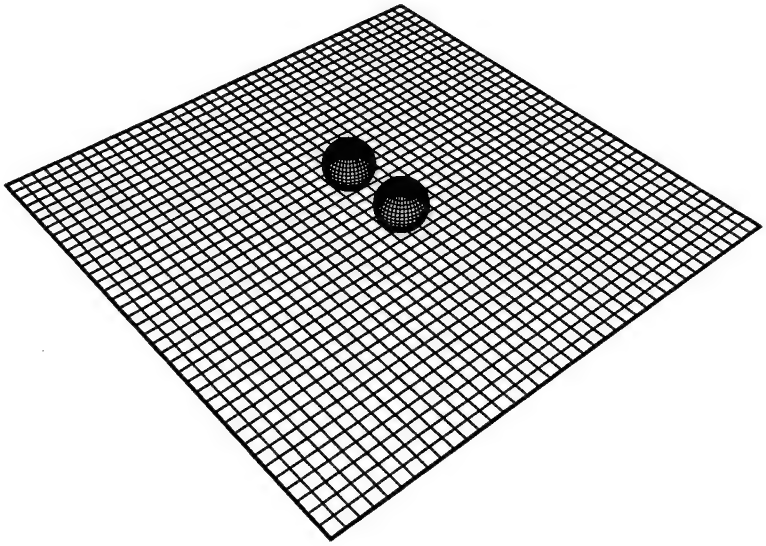


FIGURE 6-3: The two sphere problem with discretized interface. Actual discretization is shown.

Example	Code	Setup	Solve	Memory	Products	MVP Time
groundplane	FFT-LM	4.2	4.8	6.1	24	0.20
	FFT-DI	8.4	30.5	21.9	76	0.40
	FMM-DI	31.8	52.1	31.9	76	0.69
symmetry plane	FFT-LM	4.2	4.9	6.1	24	0.20
	FFT	4.4	18.6	11.4	60	0.31
	FMM	17.6	32.4	26.4	62	0.52
dielectric interface	FFT-LM	4.2	4.4	6.1	22	0.20
$\epsilon_+ = 1, \epsilon_- = 3$	FMM-DI	15.2	50.2	47.3	66	0.76
dielectric interface	FFT-LM	4.2	4.8	6.1	24	0.4
$\epsilon_+ = 3, \epsilon_- = 1$	FMM-DI	15.1	48.5	47.2	64	0.76

Table 6-3: Two sphere example. Table gives computational resources required by the layered media precorrected-FFT code [FFT-LM] and the free-space precorrected-FFT [FFT-FS] and fast-multipole [FMM] codes. The free space codes use discretized interfaces to account for the planar portion of the geometry, except for the symmetry plane case, which is implemented by duplication of the conductor discretiztion.

ϵ_+	ϵ_-	C_S	C_S^{DI}	C_M	C_M^{DI}	MVP	MVP^{DI}
1	2	146.7	146.5	-42.46	-42.45	22	60
1	4	172.8	172.4	-40.47	-40.35	22	70
1	8	201.2	200.5	-36.46	-36.12	22	72
1	16	226.3	225.2	-31.38	-30.72	22	73
1	32	244.6	243.4	-26.81	-25.83	22	74
1	64	256.0	254.7	-23.56	-22.39	24	71
1	1000	268.7	267.2	-19.67	-18.23	24	93
1	10000	269.5	268.1	-19.40	-17.93	24	X
2	1	227.3	227.5	-85.00	-84.95	24	60
4	1	423.5	424.2	-168.2	-168.1	24	62
8	1	812.4	814.4	-333.9	-333.7	24	64
16	1	1588	1593	-664.6	-664.2	24	66
32	1	3139	3148	-1326	-1325	24	66
64	1	6241	6259	-2648	-2647	24	68
1000	1	9695	9724	-4132	-4130	24	82
10000	1	96950	97220	-41310	-41270	24	291

Table 6-4: Mutual (C_M) and self (C_S) capacitance, in pF, of spheres in dielectric half-spaces, with varying dielectric constants. Spheres are of radius 1 with centers located 1.05 above interface. X indicates the non-preconditioned GMRES solver did not converge in available memory. Superscript "DI" indicates capacitances calculated using the equivalent charge formulation with discretized interfaces.

that for $\epsilon_+ \gg \epsilon_-$, the discretized interface approach generates matrices which appear to become ill-conditioned. In fact for the largest dielectric ratio considered, the iterative solver did not converge without use of a preconditioner. The deviation in the calculated capacitances is also observed to increase as the ratio of dielectric constants becomes very high. At the largest ratio considered, the relative deviation in the mutual capacitance is nearly 8%. From the data in this table, there appears to be no concern for practical problems, but there could be an indication of an underlying numerical difficulty.

A more pathological example was examined to determine if use of the layered media Green function can give more accurate results than discretizing the interface. Instead of two spheres over a ground plane, consider the case of two hemispheres, as shown in Figure 6-4. The spheres each rest on the interface of a half-space. As these are open surfaces with edges, it is in fact not clear whether the capacitances are bounded as the discretization is refined, and indeed to obtain indications of convergence, much finer discretizations of the hemispheres were needed than for the spheres. About 10000 panels were used for the ground plane discretization, and each hemisphere was discretized into 2600 panels, which corresponds to panels roughly six times smaller than for the sphere example.

Table 6-5 shows the computed capacitances for the hemispheres in dielectric half-spaces,

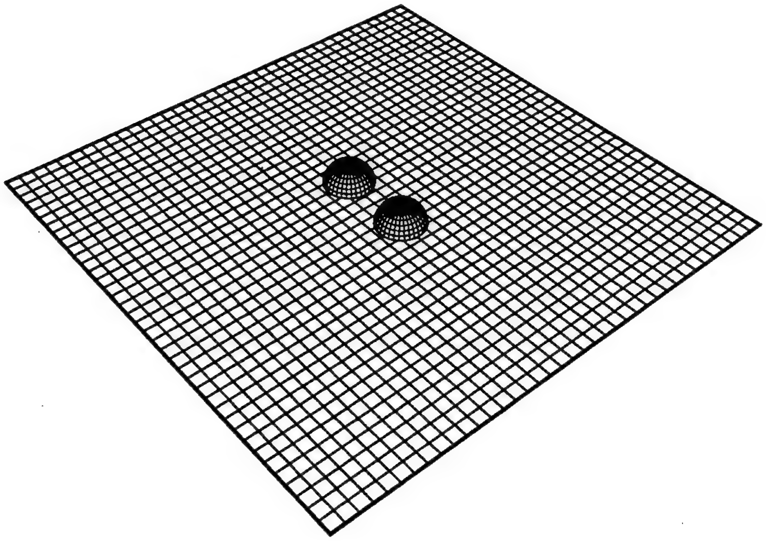


FIGURE 6-4: The two hemisphere problem with discretized interface. Actual discretization is finer than shown.

analogous to Table 6-4. Table 6-6 gives the computational performance figures for the various test cases. When $\epsilon_+ < \epsilon_-$, both codes give very similar results, which leads us to believe that both give accurate answers. However, for the reverse case $\epsilon_+ > \epsilon_-$, the free-space code with discretized interfaces gives capacitances which increasingly deviate from those calculated by the layered-media code. Note that as $\epsilon_- \to \infty$ the capacitances calculated by the layered-media code appear to be converging to those for the case of two hemispheres over an infinite ground plane of $C_S = 293.3pf, C_M = -5.83pf$. C_M calculated by discretizing the dielectric interfaces is clearly in error. Note also that nonconvergence is reached for a lower value of dielectric ratio than for the sphere example. Experiments were performed with the sphere example, with the spheres very close to the ground plane, and no significant qualitative differences from the results above were obtained. This leads us to believe that the singular nature of the charge distribution at the hemisphere edges is interacting with a difficulty in the integral formulation for the dielectric interfaces in such a way as to make the calculation inaccurate. The precise nature of the difficulty is unclear, but it may have to do with the generation of a non-trivial nullspace in the potential coefficient matrix as $\epsilon_- \to \infty$ [58, 59].

Finally we consider some more realistic geometries and some multi-layer structures. The first is the air bridge structure[15] shown in Figure 6-5. In this structure a groundplane is present at $z = 0$cm and a dielectric interface present at $z = 0.5$cm. To generate an example with panels in multiple dielectric layers, we have augmented this problem by adding a second trace, parallel to the bottom trace, at $z = 0.5833$cm, which lies in the uppermost dielectric layer. We also in this section quote results for the comb structure, for which a realistically meshed ground plane is available and included in the FMM (FASTCAP) computations. A two hemisphere example, with the hemispheres lying on a groundplane, in a dielectric layer of dielectric constant 5, with

ϵ_+	ϵ_-	C_S	C_S^{DI}	C_M	C_M^{DI}	MVP	MVP^{DI}
1	2	122.7	122.6	-29.92	-29.81	48	113
1	4	156.6	156.5	-30.76	-30.35	50	128
1	8	195.1	195.1	-28.24	-27.18	53	138
1	16	230.2	230.9	-22.78	-20.7	56	146
1	32	256.4	258.1	-16.82	-13.6	56	153
1	64	273.1	275.6	-12.21	-8.07	56	156
1	1000	291.9	295.5	-6.31	-0.92	56	X
2	1	164.8	164.9	-50.35	-50.39	44	100
4	1	294.0	294.3	-94.29	-94.41	40	101
8	1	549.6	550.3	-181.0	-181.3	38	98
16	1	1059	1061	-353.8	-354.3	36	96
32	1	2078	2081	-699.0	-700	36	100
64	1	4115	4121	-1389	-1391	35	100

Table 6-5: Mutual (C_M) and self (C_S) capacitance, in pF, of hemispheres in dielectric half-spaces, with varying dielectric constants. Superscript "DI" indicates capacitances calculated using the equivalent charge formulation with discretized interfaces. X indicates the non-preconditioned GMRES solver did not converge in available memory.

Example	Code	Setup	Solve	Memory	Products	MVP Time
groundplane	FFT-LM	18.6	48.0	23.3	56	0.85
	FFT-DI	45.2	216.1	118.0	102	2.12
	FMM-DI	50.8	279.3	137.8	103	2.71
symmetry plane	FFT-LM	18.7	28.9	22.9	34	0.85
	FFT	14.9	32.9	37.8	26	1.27
	FMM	28.2	44.3	88.5	28	1.58
dielectric interface $\epsilon_+ = 1, \epsilon_- = 3$	FFT-LM	18.9	42.7	23.2	50	0.85
	FMM-DI	(63)	(608)	299.0	124	4.9
dielectric interface $\epsilon_+ = 3, \epsilon_- = 1$	FFT-LM	18.9	34.2	23.0	40	0.86
	FMM-DI	(63)	(493)	296.0	102	4.83

Table 6-6: Two hemisphere example. Table gives computational resources required by the layered media precorrected-FFT code [FFT-LM] and the free-space precorrected-FFT [FFT-FS] and fast-multipole [FMM] codes. The free space codes use discretized interfaces.

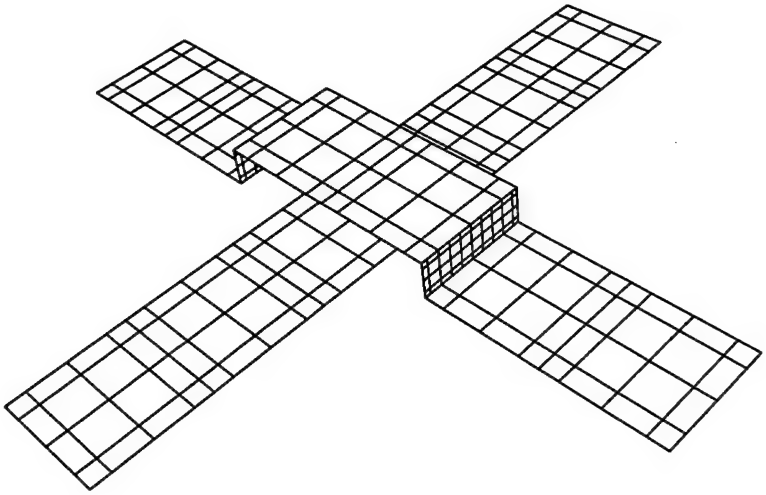


FIGURE 6-5: The airbridge example. Actual discretization is finer than shown.

a second dielectric layer, dielectric constant 1, above, is also considered.

In general, the results are in accord with what was discussed above: for the single-interface problems, the precorrected-FFT method is significantly more efficient than the fast multipole method applied to an equivalent charge (discretized-interface) formulation.

However, when the second trace is added to the air bridge example, the layered-media precorrected-FFT code becomes much less efficient. It requires four times fewer iterations to converge than FASTCAP (and note that this advantage in realistic applications would likely be ameliorated by use of a preconditioner), but has only a factor of two advantage in matrix-vector product time. In fact, when the second trace is added to the air bridge example, the matrix-vector product time increases by over a factor of two, even though the number of panels increases by only about 30%. This is due to the inefficient decomposition of space imposed by the use of the FFT. It is encouraging, however, that the setup time of the precorrected-FFT method did not increase greatly (only about 50%), despite the use of a much more complicated Green function. It is also worth noting that the precorrected-FFT method still used significantly less memory.

In summary, in this chapter we demonstrated that the precorrected-FFT method is an effective approach for electrostatic analysis of conducting bodies in stratified media. Despite becoming much less efficient when conductors are located in multiple dielectric layers, the precorrected-FFT approach is still attractive, as in many cases it is still more efficient than the equivalent-charge formulations, particularly in terms of memory utilization. The qualitative advantages of not having to generate interface discretizations are even more compelling.

Example[m,n]	Code	Setup	Solve	Memory	Products	MVP Time
comb[2,13024]	FFT-LM	65.5	78.2	60.4	51	1.53
	FMM	70.3	258.3	211	77	3.36
two hemispheres[2,5200]	FFT-LM	33.3	39.4	23.7	75	0.53
$\epsilon_1 = 1, \epsilon_0 = 4$, groundplane	FMM	58.9	709.7	531.0	194	3.67
air bridge[2,1120]	FFT-LM	8.5	5.72	4.64	31	0.18
$\epsilon_1 = 1, \epsilon_0 = 5$, groundplane	FMM	12.1	107.1	48.7	136	0.78
air bridge, second trace[3,1504]	FFT-LM	12.3	21.6	10.5	51	0.42
$\epsilon_1 = 1, \epsilon_- = 5$, groundplane	FMM	13.8	168.8	54.5	197	0.86

Table 6-7: Realistic and multi-interface examples. m is the number of non-interface conductors in the problem, n the number of panels, excluding the interface.

7

Helmholtz Problems

In the previous two chapters we gave a detailed discussion of the performance, advantages, and limitations of the precorrected-FFT method as applied to problems with Laplace kernels. In fact, most of the statements made apply to any kernel that has a convolutional structure and is singular-smooth. By singular-smooth we mean a function that is smooth on any length scale as long as it is examined sufficiently far away from the origin. Examples of singular-smooth functions are $1/r, e^{-r^2}/r, 1/(r^2+a^2)$. The problem becomes much harder when treating functions that have an additional length scale, such as $\sin kr/r$. Arbitrarily far away from the origin, this function oscillates on a scale of the wavelength $\lambda = 2\pi/k$. If the approximation of the function away from the origin does not resolve oscillations on the scale of λ the computed answers to an integral equation involving this kernel will be inaccurate. The advantage of the precorrected-FFT method is that it is easier to adapt to oscillatory kernels than, for example, algorithms which rely on a multi-level spatial decomposition. What is given up is that the precorrected-FFT method can never achieve optimal computational complexity.

In this chapter we consider a simple problem to demonstrate that the precorrected-FFT method can effectively compute the matrix-vector products stemming from a boundary-element discretization of an integral equation with a kernel of the form e^{ikr}/r and its derivatives. As a model problem we consider the scalar Helmholtz equation with Dirichlet boundary conditions. This will be the oscillatory-kernel analogy of the first-kind integral equation, Eq. 1.2. We will show that, as long as the required accuracy is not large, the precorrected-FFT method can treat the oscillatory kernel case with a surprisingly small increase in computational resources relative to the static $(1/r)$ case.

7.1 Formulation and Discretization

We wish to solve

$$\nabla^2\psi(x) + k^2\psi(x) = 0 \quad x \in \mathbf{R}^3 \backslash D \tag{7.1}$$

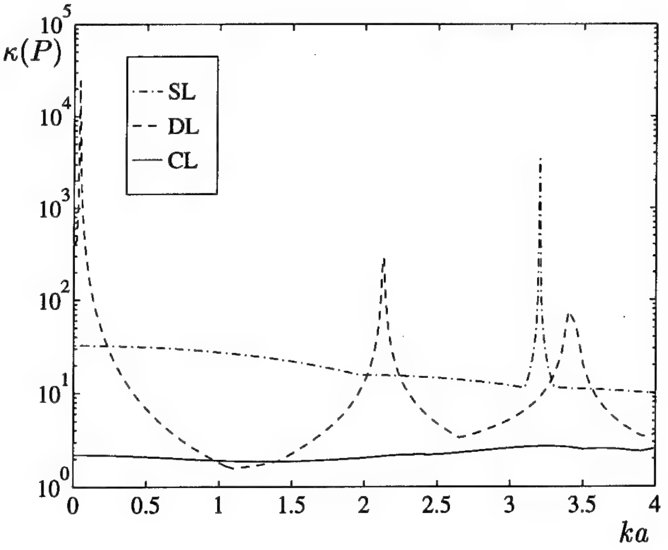


FIGURE 7-1: Condition number κ of matrix P vs. wavenumber k for 240-panel sphere, derived from integral equation based on single-layer (SL, $\eta \to \infty$), double-layer (DL, $\eta = 0$), and combined-layer(CL, $\eta = 1$) potentials.

where the domain D is a bounded region in $\mathbf{R}^3$ with boundary ∂D consisting of a finite number of disjoint closed bounded surfaces, and the exterior $\mathbf{R}^3 \backslash D$ is assumed to be connected. This problem can be cast into an integral equation form using monopole, dipole or combined-layer potentials [10]. We will consider combined-layer potentials as the produce linear system which are generally better conditioned. More importantly, integral equations derived from the combined-layer potentials do not suffer from the irregular-frequency problem as do equations derived from the plain single-layer and double-layer densities. At an irregular frequency, the integral equation ceases to have a unique solution at a set of discrete frequency values which correspond to interior resonant frequencies. This non-uniqueness of the integral equation corrupts numerical solution of the equation in the vicinity of the irregular frequency.

In the combined-layer case, the potential is represented by

$$\psi(x) = \int_S \{G_n(x, x') - i\eta G(x, x')\}\sigma(x')da', \qquad x \in S \tag{7.2}$$

where $x, x' \in \mathbf{R}^3$, S is a multiply-connected two dimensional surface in $\mathbf{R}^3$, $G(x, x') = e^{ik||x-x'||}/4\pi||x-x'||$ is the Green's function for the Helmholtz equation, G_n is the surface normal derivative of G at x', $\sigma(x')$ is the combined-layer density often referred to as a charge density, and η is a complex scalar which depends on k.

For each point $x \in \partial D$ for which $u(x)$ is specified, the charge density satifies

$$\frac{\sigma(x)}{2} + \int_S G_n(x, x')\sigma(x')da' - i\eta \int_S G(x, x')\sigma(x')da' = u(x) \tag{7.3}$$

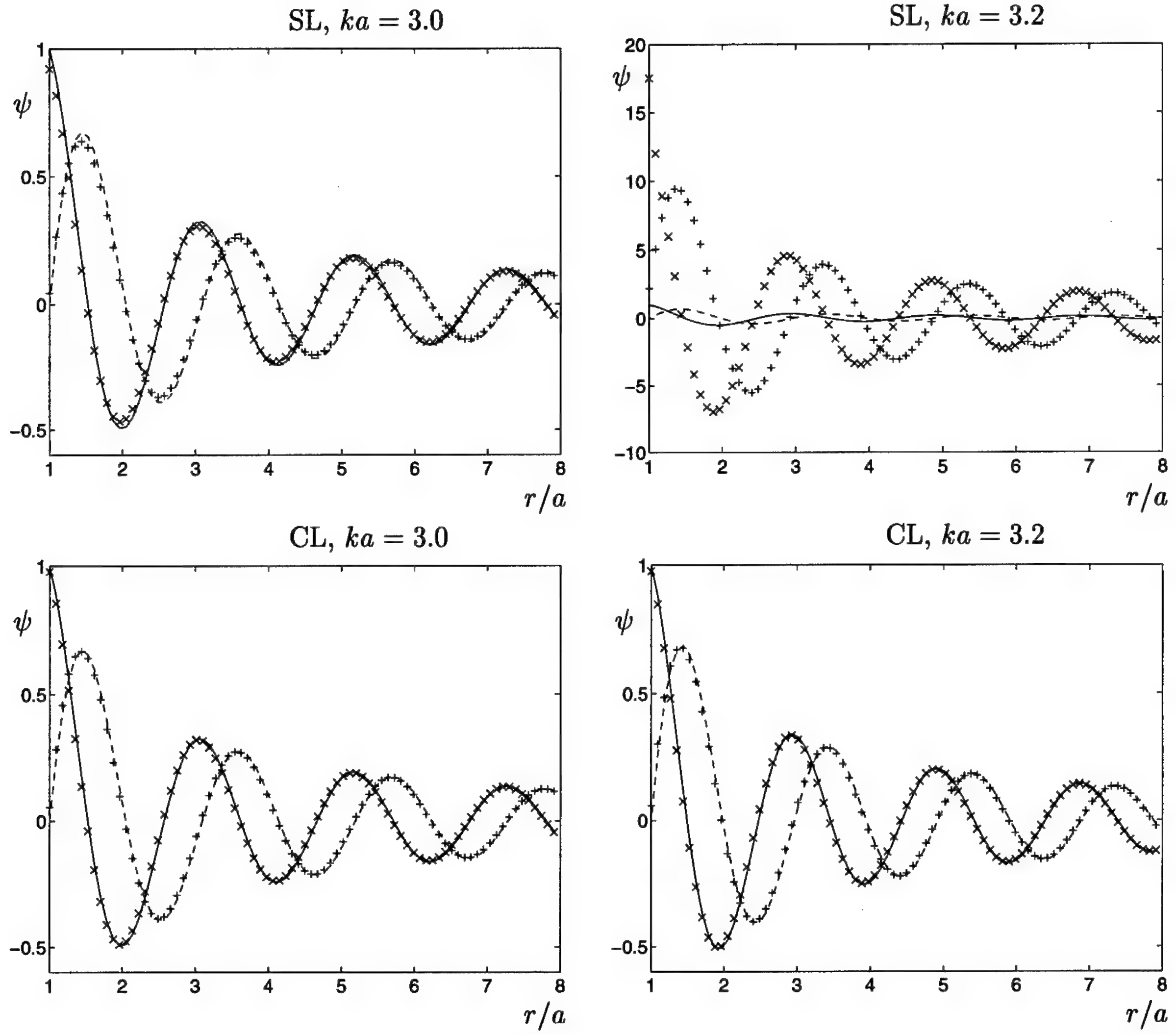


FIGURE 7-2: Effect of choice of integral equation on numerically computed scattered fields near interior resonances. Solid line: real part of exact solution. Dashed line: imaginary part of exact solution. x: computed real part of solution. +: computed imaginary part of solution. Top plots show single-layer potential, bottom plots show combined-layer potentials, at $ka = 3.0$ and $ka = 3.2$. The single-layer solution is inaccurate around the resonance $ka = \pi$, but accurate elsewhere. The combined-layer solution is accurate near both away from the resonance and near it. No matrix approximations were performed for these computations.

and for each point x where $u_n(x)$ is specified the charge density satisfies

$$\frac{\partial}{\partial n(x)} \int_S G_{n'}(x,x')\sigma(x')da' + i\eta\frac{\sigma(x)}{2} - i\eta\frac{\partial}{\partial n(x)} \int_S G(x,x')\sigma(x')da' = u_n(x). \tag{7.4}$$

In this chapter we will only consider problems of the form of Eq. 7.3 (Dirichlet problems).

To discretize Eq. 7.3 we use a similar discretization scheme as was used for the first-kind integral equations of Chapters 1-6. The charge density sigma is expanded in basis functions

$$\sigma(x) \approx \sum_{i=1}^{n} q_i \theta_i(x), \tag{7.5}$$

where the θ_i are piecewise-constant on triangular or quadrilateral panels. That is, the surface is tiled with polygons, and the θ_i are given by

$$\begin{aligned} \theta_i(x) &= 1 \qquad x \in \text{polygon } i \\ \theta_i(x) &= 0 \qquad \text{otherwise.} \end{aligned} \tag{7.6}$$

A discrete set of equations is obtained by enforcing a collocation condition to produce the linear system

$$Pq = u \tag{7.7}$$

for the panels charges q, where the matrix elements P_{ij} are given by

$$P_{ij} = \frac{1}{2} + \int_S G_n(x_i,x')\theta_j(x')da' - i\eta \int_S G(x_i,x')\theta_j(x')da', \tag{7.8}$$

$u_i = u(x_i)$, and x_i refers to the centroid of polygon i.

We emphasize that the irregular-frequency problem is due to the choice of integral-equation and is not related to any matrix approximations, such as made by the precorrected-FFT method. At any frequency, the solution to the exterior Dirichlet problem *does* have a unique solution[10]. At an irregular frequency, the integral equation itself possesses a non-unique solution, as any multiple of an interior resonant solution lies in the nullspace of the integral operator. This interior resonant solution, however, does not generate scattered (exterior) fields. When an approximate discretization is introduced, however, the various integral operators may be poorly approximated such that numerical solutions near the resonant frequency become inaccurate. Additional approximations such as made by the precorrected-FFT method will exacerbate any such numerical error. Thus when using matrix-sparsification techniques we must be particularly careful to only apply the methods to well-posed integral equations.

To see the importance of eliminating the interior resonance effects present in integral equations derived from pure single-layer and double-layer potentials, consider Figures 7-1 and 7-2. A sphere of radius a was coarsely discretized (240 triangular panels) in order to solve the boundary-value problem $\psi(a) = 1$. This has solution $\psi = ae^{ik(r-a)}/r$. The singularity of the

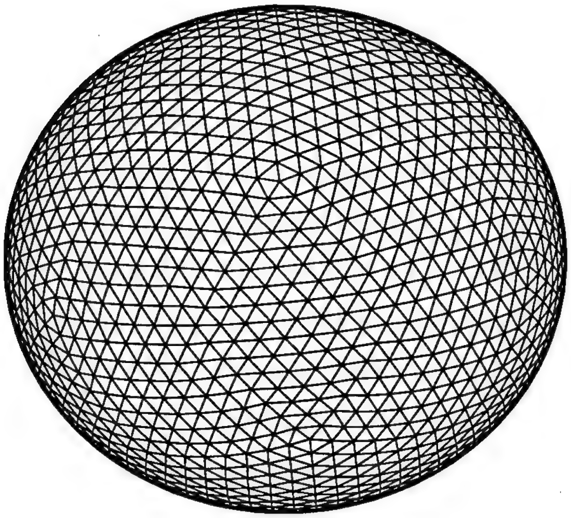


FIGURE 7-3: The 3840 panel discretization of the sphere.

integral operator is often manifested in ill-conditioning of the potential coefficient matrix P. Fig. 7-1 shows the condition number κ of this matrix for potential coefficients derived from the single-layer, double-layer, and combined-layer potentials. We see that there are apparently several internal resonances in the interval $k \in [0, 4]$. In particular, the single-layer equation has resonances where $j_n(ka) = 0$ for $n = 0, 1, \ldots$, the first of which occurs at $ka = \pi$. Therefore we expect the numerically computed solution to be inaccurate near $ka = \pi$. This is indeed the case, as is demonstrated in Fig. 7-2 in the upper-right plot. When ka is away from π, the single-layer equation gives accurate answers (at least, as accurate as can be expected given the coarse discretization). At $ka = 3.2$, however, the computed fields are grossly inaccurate. On the other hand, the combined-layer potential solution is accurate at all values of ka.

7.2 Computational Examples

Now we demonstrate that the precorrected-FFT technique can accurately compute solutions of integral equations with an oscillatory kernel. Again assume a sphere of radius a, but now with a more complicated boundary condition. In particular, let the boundary value u be given by a plane wave:

$$u = -e^{ikz}. \tag{7.9}$$

Discretizations of the sphere were generated by recursively subdividing the spherical triangles produced by the vertices of 8,20 and 60-face polyhedra. Fig. 7-3 shows a 3840 panel discretization. Figure 7-4 shows the relative average errors for four different wavenumbers, as a function of $1/n$, where n is the number of panels in the discretization. As in Figure 5-1, the errors decrease as the discretization is refined, reaching an asymptotic value at the errors induced by the grid approximation. For any discretization, the higher the frequency, the less accurate the

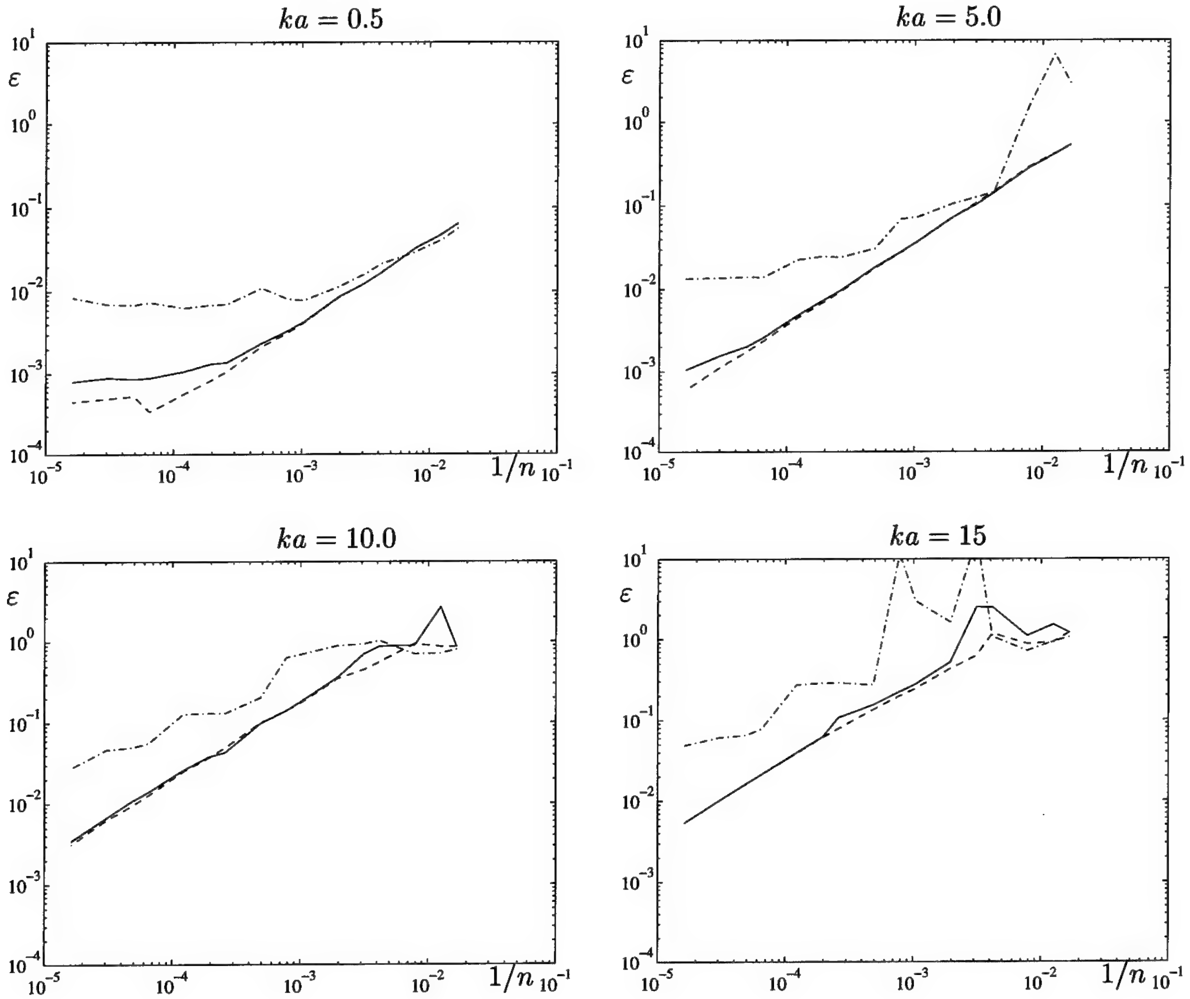


FIGURE 7-4: Errors in relative charge density for plane-wave excitation, vs. wavenumber and $1/n$, n = number of panels in discretization. Solid line: $p = 3$, dash line: $p = 4$, dash-dot line: $p = 2$.

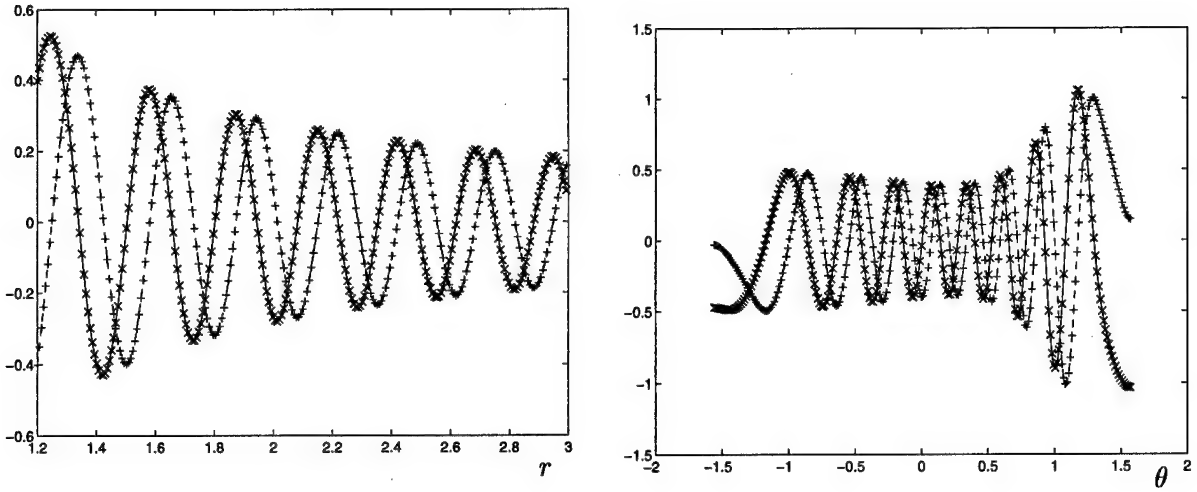


FIGURE 7-5: Scattered fields for plane-wave boundary condition on sphere, $ka = 25$. Solid line: real part of exact solution. Dashed line: imaginary part of exact solution. x: computed real part of solution. +: computed imaginary part of solution.

sphere discretization is, and so the errors are higher. Note that the asymptotic errors for a given grid order will be roughly the same as for the Laplace case (Figure 5-1), since as $n \to \infty$, $k\Delta$ where Δ is the cell size will go to zero.

To demonstrate the nature of the field solutions at higher frequencies, we take $ka = 25$, corresponding to a sphere about 8 wavelengths in diameter. The sphere was discretized into 61440 panels. Figure 7-5 shows the computed fields. The matrix solution phase of the computation required about 10 minutes on a DEC 600/333 and used 740MB of memory. About 13 CPU seconds are needed for a matrix-vector product. The average relative error in the computed combined-layer density was about 2%, and the maximum error in the computed fields was likewise a few percent. $p = 4$ grid approximations, with $k\Delta = 2.4$, where Δ is the cell size, were needed for accurate answers. Even though in realistic applications such a high frequency problem would be better treated by higher-order discretizations than piecewise-constant collocation, this example illustrates that it is possible to solve large-scale problems with relatively small computation times. The memory requirements, however, are still fairly large.

For purposes of comparison, solving the Laplace equation, using a single-layer potential, on the same geometry using precorrected-FFT accelerated GMRES, with $p = 3$, requires about 32 seconds, with convergence occuring in seven iterations and requires 215 MB of memory. The matrix vector product time is about 4.6 seconds. Note that the penalty in matrix-vector product time for switching from the real-valued single-layer $1/r$ potential to the complex valued combined-layer potential, with higher order, is only a factor of 2.8! This is less than the factor of four penalty induced by switching from direct multiplication of a real matrix times a real vector to a complex matrix times a complex vector. This is possible because the FFT of a complex sequence requires somewhat less than twice as much time to compute as the FFT of a

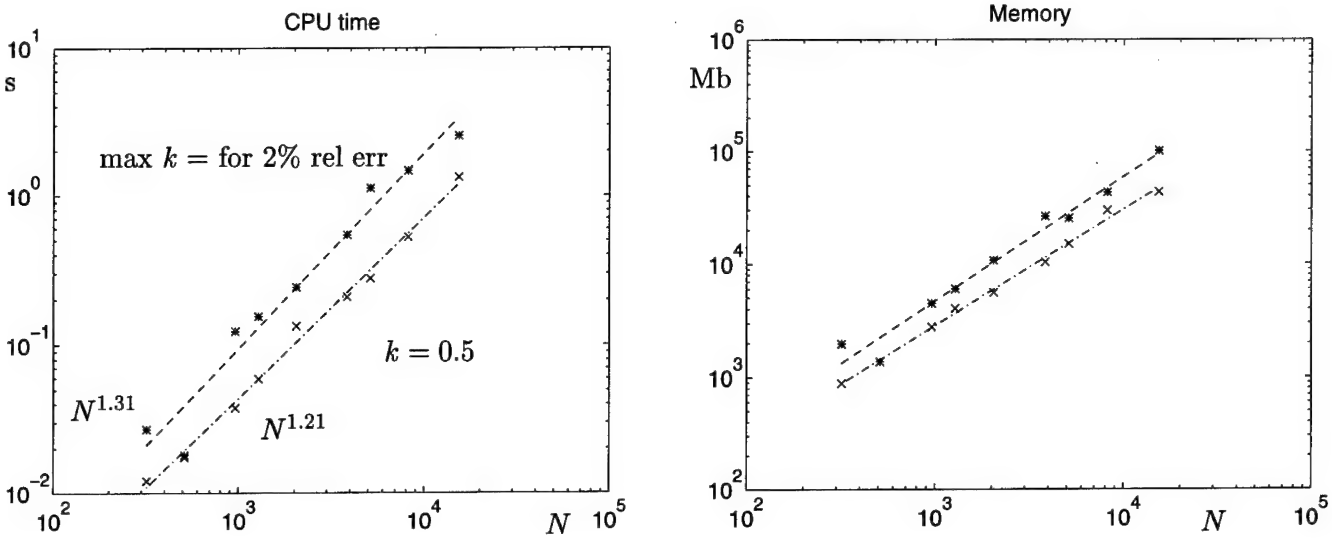


FIGURE 7-6: Computational resources needed to solve the combined-layer potential integral equation on sphere of radius $a = 1$. (x) Fixed-k experiment. (*) Variable-k experiment. (–) Linefit to variable-k points. (-.) Linefit to $k = 0.5$ points.

real sequence. Thus we expect the relative performance benefit of using the precorrected-FFT method over standard direct techniques to *increase* for the Helmholtz-potential case compared to the Laplace-case. A Gaussian elimination solution of this problem would have required about a month of CPU time on a 200MFLOPS workstation, and about 56GB of storage.

Next we analyze the behavior of the precorrected-FFT method as a function of problem size, for Laplace and Helmholtz kernels. Figure 7-6 presents the primary experimental results of this chapter. Two experiments were conducted and the CPU time required for a matrix vector product, in seconds on a DEC AXP 3000/900, and memory in MB necessary for the computations, are shown in the figure, as a function of the number of panels N in the discretized system. In both experiments we seek to obtain 2% relative mean-square error in the combined-layer charge density (which can be analytically computed). The 2% error level was selected because it allows the largest dynamic range for the limited set of discretized spheres available; in any event this choice of parameter is in principle arbitrary.

The first experiment, the results of which are shown in the lower curve marked $ka = 0.5$, involved solving the combined-layer integral equation, for the plane-wave boundary condition, at fixed $ka = 0.5$ for each of the available n-panel sphere discretizations. As $ka = 0.5$ was small for each of the n, 2% accuracy was generally obtained with grid order $p = 2$. From the dash-dot

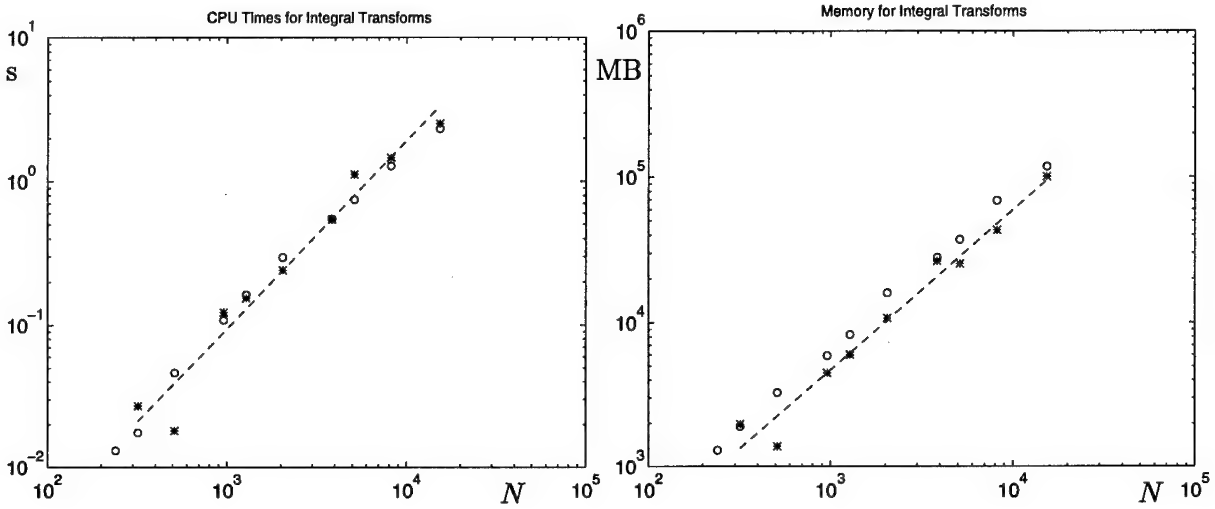


FIGURE 7-7: (o) Resources required for FMM, $l = 2$, to solve Eq. 1.2. (*) variable-k experiment from Fig. 7-6 Left figure shows CPU time needed for matrix-vector product, right figure shows needed memory, in MB.

line which is a fit to the experimental results, we see that, as predicted in Chapter 4, the CPU time and memory needed for the matrix-vector product appears to grow about as $n^{1.2}$ for fixed ka.

The second experiment, shown in the upper plots, is more difficult to explain. For each available n-panel discretization of the sphere, the wavenumber k was increased until the maximum k for which 2% relative error could be achieved was reached. Then, the precorrected-FFT parameters (p and grid spacing) were optimized to achieve minimal time-memory product subject to the 2% error requirement. As for each discretization this results in $ka > 0.5$, higher order approximations and thus more computational resources were required compared for this variable-k experiment relative to the first fixed-k experiment. From Chapter 4 we expect the time required for a matrix-vector product to grow as $n^{4/3}$ for moderate frequencies, and this is indeed what is observed experimentally. At the largest k considered on these plots, the sphere was more than four wavelengths in diameter.

It is difficult to make a comparison of the precorrected-FFT method, applied to the Helmholtz kernel, to competing algorithms, as there is a dearth of publicly available codes. Instead, in Figure 7-7 we have made a comparison to the fast multipole algorithm for the Laplace kernel. The rationale is that any fast-multipole type algorithm for Helmholtz kernels is likely to require more significant computational resources than a similar fast-multipole algorithm for the Laplace kernel, so if the precorrected-FFT method, *applied to the Helmholtz kernel* compares well with the Laplace-FMM, surely the Helmholtz-precorrected-FFT method will compare well with a Helmholtz-FMM. In Figure 7-7 we compare the variable-k results from the previous figure to the FMM, using second-order multipole expansions. In general, we see that the precorrected-FFT method can compute a matrix-vector product for a problem with an oscillatory kernel in about the same time needed by the fast-multipole method for a problem with a $1/r$ kernel.

Furthermore, despite the increased memory requirements of the precorrected-FFT method in the Helmholtz case, for all the examples considered the precorrected-FFT method required *less* memory than the Laplace-fast-multipole method executed on the same geometry.

To gain some further understanding of the efficiency of the pre-corrected FFT method at non-zero frequencies, consider Table 7-1, which shows statistics for solution of the Helmholtz equation on the 61440-panel sphere. At low frequencies, up to about five wavelengths per sphere diameter, the memory usage and CPU time required for a matrix-vector product increases only slightly over the static code (primarily, the increase reflects the differences between real and complex arithmetic, with associated storage implications). As the frequency increases to the point where the problem domain encompasses several wavelengths, however, the code starts to slow significantly. Primarily this is due to a jump in the required grid order, from $p = 3$ to $p = 4$ around the six wavelengths/diameter point. Clearly what is needed is a way to smoothly increase accuracy of the approximation without the large jumps in computation time required by changing grid order. One possibility is to include more panels in the direct interaction list. Another is to modify the FFT code, as at present it only allows array sizes which are powers of two. It is also apparent that, even when using the well-conditioned combined-layer formulation, some sort of preconditioner for the iterative scheme would be useful higher frequencies, as the tendency for the number of iterations to increase with wavenumber indicates the linear system is becoming ill-conditioned.

Finally, we compare the accuracy of the grid-collocation operators developed in Chapter 2 to operators derived from polynomial interpolation. In Chapter 5, we showed that for $k = 0$, for equivalent costs in time and memory, the grid-collocation operators produced more accurate results. Generally in numerical computations it is possible to trade accuracy for computational cost, so in this section it is instructive to consider the converse comparison, i.e., to compare the computational costs of the two schemes for a fixed error metric. Figure 7-8 demonstrates that for the fixed error ceiling of 1% relative in the charge density, the grid-collocation method is somewhat more efficient. Chapter 4 predicts that the algorithm when using polynomial interpolation should show a larger rate of growth than when using grid-collocation. While the linefit to the polynomial data does have a larger slope, the available data does not seem sufficient to make any firm claims about complexity.

d/λ	MV-product time	Memory	Iterations	Approximation
0 (FASTCAP)	1.14	2.30	10	$l = 2$
0 (FFT)	1.0	1.0	7	$p = 3$
0.5	1.96	1.80	6	$p = 3$
1.0	1.96	1.80	6	$p = 3$
3.0	2.00	1.85	17	$p = 3$
5.0	2.06	1.91	32	$p = 3$
7.0	2.83	3.49	44	$p = 4$

Table 7-1: Resources required to solve scalar Helmholtz equation on a sphere of diameter d discretized into 61440 panels. Memory and times are normalized to those required for the static precorrected-FFT code. Parameters chosen for 1% relative error in charge density. More iterations are needed for $ka = 0$ than $ka = 0.5$ because the $ka = 0$ codes use the less well-conditioned single-layer formulation.

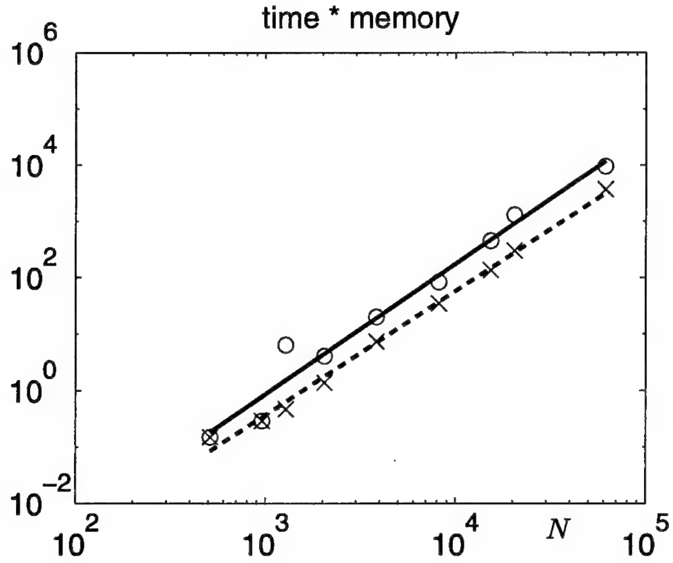


FIGURE 7-8: Product of cpu time-memory for matrix-vector products for 1% error solution of Helmholtz equation on sphere.(o) polynomial interpolation operators, (x) grid-collocation operators.

8

Accelerated full-wave EM analysis

Many aspects of engineering design, such as interconnect delay estimation, signal integrity analysis, and electromagnetic compatibility applications, require understanding the electrical properties of complex structures. However, because of the great computational cost, full-wave simulation of large, complex three-dimensional structures is rarely used in the engineering design and optimization process. At best, a range of representative structures is simulated, and/or a variety of engineering approximations are made to derive either analytical models, less computationally demanding numerical models, or sets of rule-of-thumb design guidelines. While these approaches are simple and intuitive, as interconnect technologies become more complex and signal frequencies rise, the task of obtaining simple yet accurate models of geometrically complex structures will become more difficult, and time spent deriving and applying simple models may become burdensome. Thus, efficient, fully three-dimensional, full-wave simulation tools will be increasingly needed for engineering design of microelectronic systems.

Finite-difference methods, particularly the finite-difference time-domain (FDTD) technique, and finite-element methods are popular approaches to full-wave electromagnetic analysis. These difference-based schemes generate sparse, but very large, models. In contrast, methods based on integral equation formulations, such as method-of-moments techniques, generate smaller, but dense, matrix representations. While the difference-based methods can operate in the time or frequency domain, integral-equation based codes are usually frequency-domain. Development of integral-equation based tools has been hampered by the high computational complexity of dense matrix representations and difficulty in obtaining and utilizing the frequency-domain response.

In section 8.1 we review the retarded partial-element equivalent circuit (rPEEC) electromagnetic formulation[63] used for full-wave electromagnetic analysis. We emphasize, however, that our numerical methods are not tied to this formulation. Section 8.2 reviews the precorrected-FFT algorithm used to construct a sparse approximate representation of the rPEEC model and introduces notation which will be useful in Chapter 9.

8.1 The rPEEC formulation

We begin by expressing the electric field in terms of the vector potential $\vec{A}$ and the scalar potential ϕ

$$\vec{E} = -\frac{\partial \vec{A}}{\partial t} - \nabla\phi \tag{8.1}$$

which after a Laplace transformation becomes

$$\vec{E} = -s\vec{A} - \nabla\phi \tag{8.2}$$

In the Lorentz gauge, the potentials in Laplace domain are related to the currents $\vec{J}$ and charges q by:

$$\vec{A}(x,s) = \frac{\mu}{4\pi}\int d^3x'\vec{J}(x',s)\frac{e^{-s|x-x'|/c}}{|x-x'|} \tag{8.3}$$

$$\phi(x,t) = \frac{1}{4\pi\epsilon}\int d^3x' q(x',s)\frac{e^{-s|x-x'|/c}}{|x-x'|} \tag{8.4}$$

The exponential factor in the integrals of Eq. 8.3, 8.4 represents a delay (retardation) in the time-domain which is due to the finite propagation speed of light.

Assuming a set of conductors in a uniform medium, in every conductor, Ohm's law $\vec{E} = \vec{J}/\sigma$, with σ the conductivity, allows us to write

$$\frac{\vec{J}(x,t)}{\sigma(x)} + s\vec{A}(x,t) + \nabla\phi = 0 \tag{8.5}$$

often known as the electric field integral equation (EFIE). Adding the continuity equation

$$\nabla \cdot \vec{J}(x,t) + sq(x,t) = 0 \tag{8.6}$$

to Eq. 8.5 produces a system of equations which, given appropriate boundary conditions, can be solved for the charges q on conductor surfaces and the currents $\vec{J}$ flowing in the conductors.

To achieve a method-of-moments[1] discretization of the problem, rectangular elements of constant current are used to represent $\vec{J}$, and the charges are assumed to be piecewise-constant over rectangles which tile the conductor surfaces. After discretization, the charges are eliminated as variables, leading to the single equation for the vector of discretized currents $j(s)$

$$\left[T^T P(s) T + sR + s^2 L(s)\right] j(s) = su \tag{8.7}$$

with T the incidence matrix for the equivalent circuit. In Eq. 8.7, $P(s)$ is a dense matrix coupling all the capacitive cells (it represents the scalar potential, or "capacitive" interactions). Its elements are

$$P_{ij}(s) = \frac{1}{4\pi\epsilon_0}\int_{x'\in S_i}\int_{x\in S_j} d^2x' d^2x \frac{e^{-s|x-x'|/c}}{|x-x'|} \tag{8.8}$$

where S_i, S_j represent the i, jth rectangular charge basis function respectively. R is a diagonal matrix representing restistances. $L(s)$ represents the inductive interactions, and its matrix elements are similar to those of $P(s)$.

Eq. 8.7 can be written in terms of a single frequency dependent matrix $A(s)$ (not to be confused with $\vec{A}$)

$$A(s)j(s) = su \tag{8.9}$$

Given a vector u of voltage source excitations of the equivalent circuit, Eq. 8.9 can be solved for the discrete currents $j(s_n)$ at a particular complex frequency s_n.

8.2 Grid-based matrix sparsification

The difficulty with the rPEEC formulation, as in all integral-equation based electromagnetic solvers, is that it contains frequency-dependent dense matrices. That is, the matrices $P(s)$ and $L(s)$ have $O(n^2)$ non-zero entries, where n is the number of degrees of freedom in the discretized system, all of which must be accounted for to obtain an accurate solution. Presently, most full-wave method-of-moments solvers at some point use Gaussian elimination to perform an LU-decomposition of the matrix $A(s)$. In that case, simply storing the matrix requires $O(n^2)$ memory, and $O(n^3)$ floating-point operations. For a complex matrix of size $n = 10,000$, nearly 1.5 gigabytes of memory would be needed to store the matrix in double-precision, and a machine capable of 100 MFLOPS would take almost eight hours to perform a *single* LU-decomposition.

A more effective approach is to construct a sparse representation $\hat{A}(s)$ of the system description $A(s)$, such that multiplication of a vector y by $\hat{A}(s)$ can be performed in close to $O(n)$ operations and storage, and such that the difference $||\hat{A}(s)y - A(s)y||$, where $|| \quad ||$ denotes vector norm, is small for an arbitrary vector y. To obtain the system response $j(s)$, a Krylov-subspace iterative matrix solver such as GMRES[11] is used. Given a complex frequency s_0, such algorithms can compute the solution $j(s_0)$ to the linear system of equations $\hat{A}(s_0)j(s_0) = b$, for given b by performing only matrix-vector product operations with the matrix $\hat{A}(s_0)$. If the number of iterations needed for convergence is bounded, the resulting algorithm will need close to $O(n)$ time and storage to compute a solution at a single frequency point.

For the quasi-static (non-retarded) case, a variety of effective algorithms is available, the most well-known of which is the Fast Multipole Method [64] used in FASTCAP and FASTHENRY. The oscillatory Green's function (Eqs. 8.3, 8.4) used in full-wave electromagnetic analysis is more difficult to treat, and although fast-multipole type and multigrid algorithms have been proposed, they are not as mature as algorithms for the quasistatic case.

In the preceding chapters, a multigrid-like[6] "precorrected-FFT" algorithm was presented which for not-too-inhomogeneous geometries significantly reduces the $O(n^2)$ time and memory needed to compute a matrix-vector product. As we shall see in Chapter 9, this algorithm is of particular interest in the context of model reduction, as a single algorithm can be used to span the entire range of frequencies, including zero frequency, which is not the case for the fast multipole algorithms.

Algorithm 1 (Precorrected-FFT).

```
1. Project "charge" $q$ onto grid : $q_g = Wq$
2. Compute grid-charge potentials $\phi_g$ (FFT) : $\phi_g = U(s)q_g$
3. Interpolate grid potentials : $y_g = W^T\phi_g$
4. Add local interactions : $y = y_g + D(s)q$
```

To derive the algorithm, first consider the evaluation of the sum

$$y(x_j) = \sum_i g(|x_i - x_j|, s)q(x_i) \quad i, j = 1 \ldots N \tag{8.10}$$

for some set of N discrete "charges" q at points x_i, and evaluating the "potential" y (as given by the Green function $g(|x_i - x_j|)$) at all the other charge positions x_j. This sum may be approximated in a four step process (Algorithm 1). A uniform grid is introduced which covers the problem domain (note that the grid is not in any way linked to the underlying problem discretization). The first step is to represent the charge q on the grid. By this we mean that for each charge, a small set of point charges that lie on the grid and surround the charge being "projected" is used to approximate the long-range potential of the charge. Second, the potential of all the grid charges is computed at the grid points. This operation can be accomplished in several ways, the simplest of which is by use of the FFT.[1] Third, the potential on the grid is interpolated onto the evaluation points. Finally, since the grid representation will only be accurate far away from the charge being approximated [6, 38], the potential of nearby charges must be computed exactly.

If we introduce the auxiliary variables q_g for the grid charges and ϕ_g for the grid potentials, we can write the precorrected-FFT algorithm as

$$q_g = W(s)q \tag{8.11}$$

$$\phi_g = U(s)q_g \tag{8.12}$$

---

[1] For highly inhomogeneous problems, a multigrid algorithm with more than two grid levels could be used.

$$y = D(s)q + V(s)\phi_g \tag{8.13}$$

where $W(s)$ is a (possibly frequency-dependent) projection operator, $U(s)$ is a Toeplitz matrix which corresponds to convolution of the Green function with the grid charges to give grid potentials, $V(s)$ is an interpolation operator, and $D(s)$ is a sparse matrix which represents interactions between nearby charges. Thus the precorrected FFT algorithm replaces the relation

$$y = G(s)q \tag{8.14}$$

with dense matrix $G(s)$ by an approximate representation

$$\hat{y} = [D(s) + V(s)U(s)W(s)]\, q \tag{8.15}$$

$U(s)$ is dense, but due to the use of the FFT it possesses a sparse factorization, so the representation of Eq. 8.15 is effectively sparse.

The problem of perfoming a multiplication by the matrix $P(s_0)$, for some s_0, is very similar. In this case the Green function is given by

$$g(|x_j - x_j|, s_0) = \frac{e^{-s_0|x_i - x_j|/c}}{|x_i - x_j|} \tag{8.16}$$

where c is the speed of light. The potential, instead of being due to point charges, is now given by a uniform charge density over the surface of a rectangle, and the potential is not just evaluated at one point, but integrated over the rectangle surface. For sufficiently separated panels, however, the integral is well approximated by the potential of a point charge at a rectangle centroid evaluated at the other rectangle's centroid. Thus, the interpolation and projection operators V, W can be taken to correspond to point charges, and so the precorrected-FFT algorithm we have used for rectangular elements differs from the point-charge case only in the way the local interactions are treated.

As will be clear later, for the purposes of model reduction it is convenient to choose the interpolation and projection operators to be frequency-independent. The simplest such choice corresponds to polynomial interpolation. As is discussed in Chapter 2 and [6, 38], the problems of projection (step 2) and interpolation (step 3) are completely equivalent. Thus, since a symmetric discretization technique was used in the rPEEC formulation, the interpolation operator V is simply the transpose of the interpolation operator W, and the sparse factorization of the $A(s)$ matrix is

$$\hat{A}(s) = D(s) + W^T U(s) W \tag{8.17}$$

It can be concluded from the results of Chapters 2 and 7 that Algorithm 1, resulting in the factorization of Eq. 8.17, calculates a product operation, including the effects of all long-range interactions, to engineering accuracy when even fairly low-order interpolation operators are used[6, 38].

9

Model Reduction Issues

Problems with oscillatory kernels, such as the Helmholtz problems discussed in Chapter 7 and the full-wave electromagnetic models presented in Chapter 8, introduce two difficulties when compared to the simple, static Laplace equation. First, the oscillatory kernel is more difficult to approximate. We have already shown that the precorrected-FFT method can overcome this difficulty. Second, often what is needed is to determine a system response as a function of a parameter such as frequency. Thus in principle many parameter-dependent integral equations must be solved, as opposed to the case of electrostatic analysis, where for a given geometry only a single integral equation need be solved (although possibly for several different right-hand-side excitations).

The simplest approach to obtaining the frequency-dependent response is to solve the relevant integral equation at a set of discrete frequencies. At each frequency the solution can be obtained via the precorrected-FFT accelerated iterative scheme discussed in the previous chapters. There are two primary difficulties with this approach. First, it is inefficient as it does not exploit the fact that the solutions are usually smoothly varying as a function of frequency. It should be possible to simultaneously obtain the integral equation solutions for two frequency points that are close together with less effort than if the solution at each of the two points were performed completely independently. Secondly, the frequency response may contain contributions from poles near the imaginary axis. Such poles generate sharp spectral features that are difficult to resolve using discrete sampling on the imaginary axis, but they can be resolved in an efficient manner using rational function approximation.

In this chapter, we introduce a hybrid algorithm which incorporates features of orthogonalized Krylov methods[65] and the series-expansion based methods[66] to construct a multipoint rational approximant, for a system with distributed elements, in a manner similar to the "complex frequency hopping" (CFH) multipoint algorithm[31]. The resulting series-Krylov (SKCFH) algorithm in combination with a multilevel transform representation yields an efficient and elegant solution of the electromagnetic problem. We demonstrate that an algorithm based on application of a novel model-order reduction scheme directly to the sparse model generated by a fast integral transform has significant advantages for frequency- and time-domain simulation.

This method can obtain a high-order rational approximation (or reduced-order model) of the sparse representation (Eq. 8.17) in a numerically stable manner.[1] The high order is necessary in order to represent the fine spectral features and retardation in the frequency response of typical electromagnetic models. Explicit moment-matching techniques are not numerically stable and will not be sufficient for this problem. Orthogonalized Krylov subspace methods, such as the nonsymmetric Lanczos algorithm or the Arnoldi method [29, 28, 65], while useful for stably constructing arbitrarily high order approximants of lumped systems, are not directly applicable to distributed (delay) systems such as transmission lines or rPEEC. We extend the techniques aforementioned to created a series-Krylov CFH (SKCFH) algorithm which is efficient (requires few expansion points), numerically stable, and applicable to model reduction of distributed and delay systems.

9.1 Rational approximation via orthogonalized Krylov methods

Before considering the generate rational approximation problem for the fullwave rPEEC model presented in Chapter 8, it is instructive to consider approximate models of lumped linear dynamical systems, as model reduction procedures for these systems are well-developed.

Consider the Laplace domain system description of a linear lumped network,

$$sAx = x + b \tag{9.1}$$

$$y = c^T x \tag{9.2}$$

where A is the matrix desciption of the system that relates the entries of the input vector b to the set of unknown system variables in the vector x, and y is the vector of outputs obtained from the internal system variables via a mapping vector c^T. It is clear that the frequency dependent response $y(s)$ of such a system will be,

$$y(s) = -c^T(I - sA)^{-1}b. \tag{9.3}$$

The response $y(s)$ is a rational function of the Laplace frequency s, so it is reasonable to consider approximating $y(s)$ via a lower order rational function. Moment matching methods use the sequence of moments $c^Tb, c^TAb, c^TA^2b \ldots c^TA^kb$ to obtain a Padé approximation to $y(s)$. In finite arithmetic, at some k (which could be small!) the vectors A^kb and $A^{k-1}b$ will no longer be linearly independent and the process breaks down. No accurate higher order approximation can then be obtained.

---

[1] Note that the use of the word "stable" in this section refers to numerical stability of the model reduction algorithm, not time-stability of the reduced-order model, which is an important but separate issue.

In order to obtain a more stable algorithm, we consider algorithms that operate explicitly in the Krylov subspace

$$\mathcal{K}\{A,b\} \equiv \{b, Ab, A^2b, \cdots\} \tag{9.4}$$

of the matrix-vector pair $\{A, b\}$. In orthogonalized Krylov-subspace approaches to model reduction, a reduced order matrix model is constructed based on a set of vectors that span the Krylov subspace $\mathcal{K}\{A,b\}$ (and possibly $\mathcal{K}\{A^T, c\}$). By retaining an orthogonality relation among the vectors, linear independence can be maintained, and so high order rational approximants can constructed.

For the model reduction problem, at least two alternative Krylov subspace algorithms exist: the nonsymmetric Lanczos algorithm and the Arnoldi method. We have chosen the Arnoldi method for several reasons. First, a rational approximant for all the variables in the system is directly and naturally generated, which is useful in, for example, electromagnetic compatibility analysis. Second, for our dense-matrix problems and low-order (compared to the system size) approximations, we do not expect the full orthogonalization needed in the Arnoldi method to be as costly as in sparse matrix problems where the nonsymmetric Lanczos method might be more efficient. Finally, our approach to distributed systems is easier to describe and implement using the Arnoldi approach.

Algorithm 2 (Arnoldi).

```
$v_1 = b/||b||$
for $k = 1, 2, \ldots q$
    $w = Av_k$
    $H_{i,k} = v_i^H w \quad i = 1, \ldots, k$
    $w = w - H_{i,k} v_i \quad i = 1, \ldots, k$
    $v_{k+1} = w/||w||$
    $H_{k+1,k} = ||w||$
end
$V_q = [v_1 \; v_2 \; \ldots \; v_q]$
```

The Arnoldi algorithm applied to the matrix pair A, b for q steps generates $q+1$ orthonormal vectors spanning the subspace $\mathcal{K}\{A,b\}$ as the q columns of the matrix V_q and the vector v_{q+1}. As a product of the orthogonalization procedure, the method produces a $q \times q$ upper Hessenberg matrix H_q and scalar $h_{q+1,q}$ which satisfy,

$$AV_q = V_q H_q + h_{q+1,q} v_{q+1} e_q^T \tag{9.5}$$

where e_q is the qth unit vector. It can be shown that a matrix rational function $G_q^A(s) = \|b\|_2 c^T V_q (I - sH_q)^{-1} e_1$ is a reduced order model of the original system, Eq. 9.3, which matches its first $q-2$ moments[65]. Generally an accurate model can be obtained with q much smaller than the system dimension. The only operations with the matrix A which are needed are matrix-vector product operations, making the method attractive for sparse systems. The difficulty with applying these algorithms to distributed systems such as those described by rPEEC is that these systems have a frequency-dependent A matrix, $A(s)$. In order to be able to apply the Arnoldi method to distributed systems such as networks with transmission lines, or rPEEC models, we must construct a first-order lumped system description.

9.2 Reduction of distributed systems

The fullwave rPEEC models are not lumped linear models as they contain delay factors of the form e^{-sT}. Consider the Laplace domain representation of a general distributed system

$$A(s)x = b \tag{9.6}$$

$$y = c^T x \tag{9.7}$$

where we have allowed the system matrix $A(s)$ to have arbitrary frequency dependencies. In general, Eq. 9.6 may describe an infinite-order linear system. That is, the Taylor expansion of the matrix operator $A(s)$ may contain infinitely many non-zero terms. To utilize the Arnoldi algorithm for model reduction, we must first convert this infinite-order, finite-dimensional system into a first-order, infinite dimensional system.

One way to obtain a first-order representation of Eq. 9.6 is to expand $A(s)$ in a Taylor series

$$A(s) = A_0 + sA_1 + \frac{s^2}{2!}A_2 + \frac{s^3}{3!}A_2 + \dots \tag{9.8}$$

The equivalence above is satisfied for all values of s if the matrix $A(s)$ is entire in the complex plane; that is, if it contains no entries with finite singularities. In that case the radius of convergence of the Taylor series is infinite. This is true in the case of, for example, rPEEC circuits where $A(s)$ contains only algebraic entries and delay factors (exponentials in the Laplace domain).

Substituting Eq. 9.8 into Eq. 9.6 and multiplying by A_0^{-1}, the system becomes

$$\left[I + s\tilde{A}_1 + \frac{s^2}{2!}\tilde{A}_2 + + \frac{s^3}{3!}\tilde{A}_3 + \dots \right] x = A_0^{-1} b \tag{9.9}$$

where $\tilde{A}_k = A_0^{-1} A_k$ and I is the unity matrix. We recursively define new variable vectors as follows,

$$x_0 \equiv x, \quad x_1 = \frac{s}{2}x_0, \quad x_2 = \frac{s}{3}x_1, \quad \cdots, \quad x_k = \frac{s}{k+1}x_k, \quad \cdots$$

The system of Eq. 9.6 becomes

$$\left\{ I - (s) \begin{bmatrix} -\tilde{A}_1 & -\tilde{A}_2 & -\tilde{A}_3 & -\tilde{A}_4 & \dots \\ I/2 & & & & \\ & I/3 & & & \\ & & I/4 & & \\ & & & \ddots & \end{bmatrix} \right\} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} A_0^{-1}b \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \tag{9.10}$$

$$y(s) = \begin{bmatrix} c^T & 0 & 0 & 0 & \dots \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \\ x_3(s) \\ x_4(s) \\ \vdots \end{bmatrix} \tag{9.11}$$

or

$$(I - s'\overline{A})\overline{x} = \overline{b} \tag{9.12}$$

$$y(s) = \overline{c}^T \overline{x}(s) \tag{9.13}$$

This is a first order system of infinite dimension which allows us to apply a Krylov-subspace based model-reduction algorithm, just as for the system of Eq. 9.6. More importantly, for finite order, the model-reduction algorithm can be executed in a finite number of steps since the starting vector, $\overline{b}$, is bottom sparse (only the first n entries are non-zero). When an Arnoldi process is used to generate a upper-Hessenberg representation of the matrix $\overline{A}$, from the structure of $\overline{A}$, each Arnoldi vector v_k will have kn non-zero entries in it. So, at order k, the number of original n-size matrix products required to obtain the next Arnoldi vector v_{k+1} will be k. Thus total storage and computational costs will be $O(q^2n)$ for an order-q model.

Now we apply this algorithm to the sparse rPEEC factorization (Eq. 8.17). Assuming a Taylor expansion about a complex frequency s_0, the matrices in the Taylor expansion of the system description are

$$A_k = D_k(s_0) + W^T U_k(s_0) W \tag{9.14}$$

where D_k and U_k are the Taylor-expansion matrices of $D(s)$ and $U(s)$. To implement the model reduction algorithm, it is necessary to compute matrix-vector products with the matrices $A_k, k > 0$, and solve linear systems of the form $A_0 x = b$ for arbitrary b. Step k of the Arnoldi model-order reduction algorithm requires k products with $A_1, \dots, A_k$ and one linear system solution with A_0. The A_k products are straightforward. Since it is possible to compute matrix-vector products with A_0, the linear systems can be solved using a Krylov-subspace iterative algorithm such as GMRES[11]. Several techniques, such as preconditioning and Krylov-subspace recycling[67, 68, 69], can be used to accelerate the convergence of the iterative algorithm, but we will not discuss them in detail here.

One principle difficulty with the proposed algorithm is that as the matrices D_k, U_k are needed for many k, either they must all be precomputed and stored, at a great increase in memory cost, or they must be computed in runtime. Generally the memory requirements are limiting, so at each order q, the information needed to compute products with $D_k, U_k, k \leq q$ must be computed. In particular, a precorrection step is needed for each D_k. This greatly increases the effective cost of a matrix-vector product relative to the single-frequency case. That is, products with the q terms $A_k, k = 1 \ldots q$ cost much more than k times the cost of a single product with A_0 (assuming as is the usual case that the information needed to represent A_0 has been precomputed). On the other hand, if only a few orders in the Taylor expansion are needed, for example at low frequencies (and regardless of model order), each of the D_k, U_k matrices can be precomputed, and the resulting combined algorithm is quite efficient.

9.3 Multipoint rational approximants

Our approach is based on a Taylor series expansion of the exponential function, and for large arguments, corresponding to frequencies far from the expansion point, it is not possible to accurately sum an arbitrary number of terms in this series. Thus, unlike Krylov methods for lumped systems, we cannot obtain accurate models of *arbitrarily* high order from a single expansion point. This is not a concern, however, as it is more efficient to obtain the rational approximant from moderate-order expansions about multiple expansion points, rather than a single high order expansion [70], particularly since the cost of the model generation grows as q^2. Our approach is similar to complex frequency hopping[31] where several explicit moment-matching expansion points (hops) are used to generate a transfer function over a broad frequency range. Once it is determined that an approximation has reached its limit in accuracy and/or CPU usage then another expansion point is chosen. From CPU considerations hops of order 20-40 seem to be optimal. Several options exist to utilize the information from multiple expansion points. A CFH-style algorithm, based on analyzing converged poles, can be used to construct a single reduced-order model by extracting the poles and residues at each expansion point and rejecting inaccurate ones. Pole convergence may be determined by identifying common poles in neighboring expansions, as in CFH, or by using estimates based on residuals of Ritz pairs, which are available directly from the Arnoldi process[28, 65]. Alternatively, the two expansion points can be deemed to be accurate in the range between them when their frequency response matches at some intermediate point, and the full frequency response obtained via piecewise-rational function interpolation using the frequency responses from the various expansion points as interpolation functions.

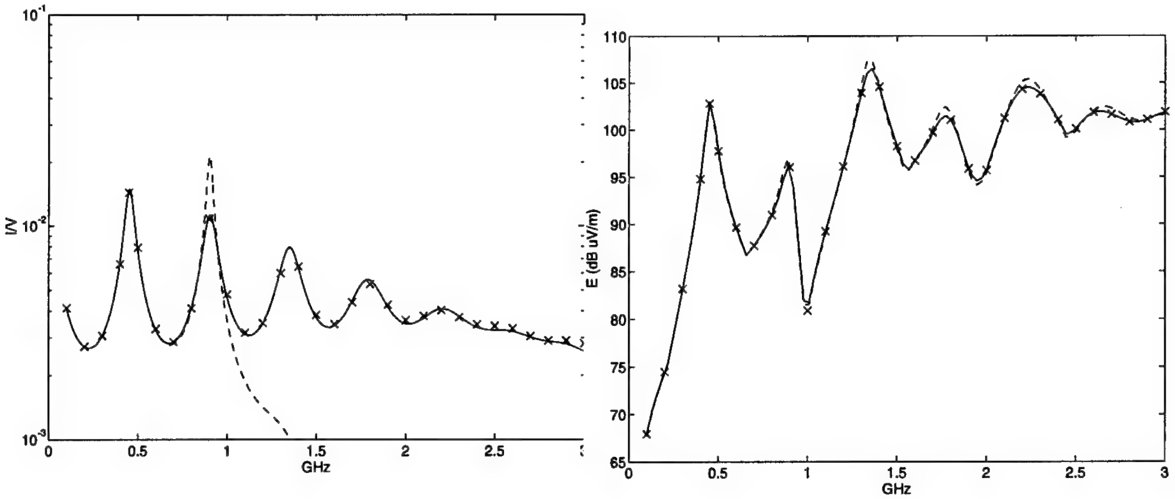


FIGURE 9-1: Frequency response of the two-strip example. Left (a): Amplitude of driven current vs. excitation frequency, 100 MHz to 3 GHz. Solid lines show directly computed response and Arnoldi-based approximation of dense model. Dashed line shows AWE approximation of dense model. 'x' shows response at selected frequency points of sparse model, computed using the iterative algorithm GMRES. Right (b): maximum amplitude of electric field at 3 meters from structure. Solid line shows reduced-order model of dense system. Dashed line shows reduced-order model of approximate sparse system. 'x' shows full dense model.

9.4 Computational Examples

We now consider two representative examples which illustrate various aspects of the model-order-reduction algorithm as applied to the sparse rPEEC representation. The first example is a simple configuration of two strip conductors, intended to illustrate the capabilities of the new model-order reduction algorithm. Two parallel 30cm long strips, 5cm apart and 1cm wide, are driven at one end by a voltage source with 50 ohm internal impedance, and are terminated at the other end by a 10 ohm resistive load. Figure 9-1 shows the magnitude of the current driven through the load by a unit voltage source, as a function of frequency.

We have calculated the response using explicit single point moment-matching, as well as the Arnoldi-based model-reduction algorithm applied to the dense rPEEC model, by solving iteratively for the frequency response using the sparse rPEEC model, and by performing matrix factorization of the dense rPEEC model. The best AWE approximant it was possible to obtain used 20 moments. Moment-matching can resolve the first resonance and detect the presence of the second, but cannot match the actual frequency response past about 0.8 GHz. On the other hand, the Arnoldi based approach was able to accurately match the frequency response over a 3GHz range using an 85th order model. We stress that this is actually an example where moment-matching performs relatively well. Examples are easily constructed[29] for which a single point moment expansion breaks down after only a few moments, in which case virtually

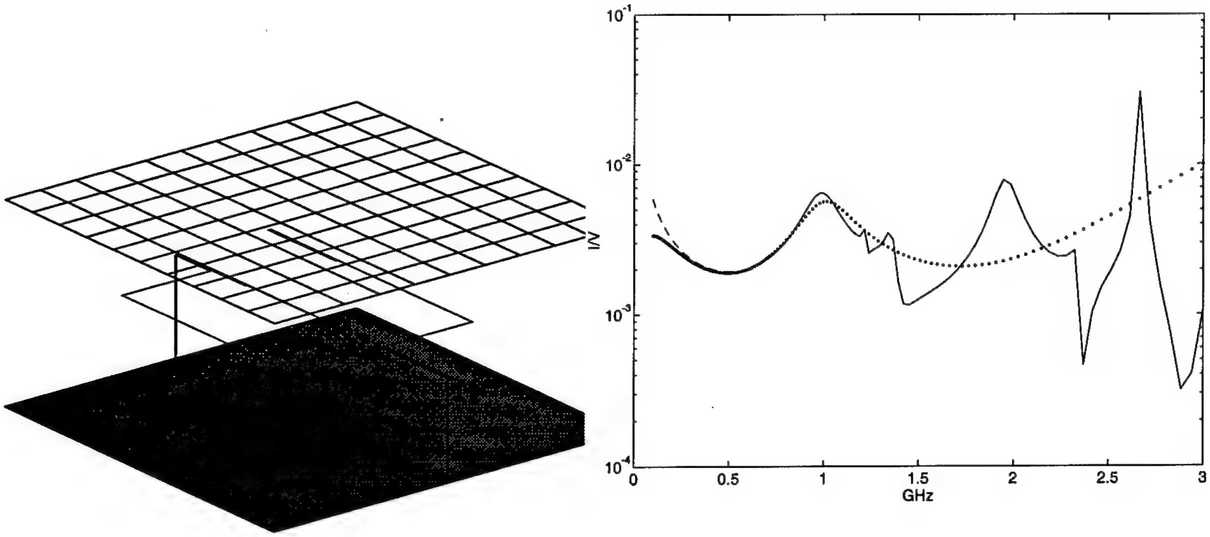


FIGURE 9-2: Left : a large interconnect structure. Vertical axis is not to scale. Right: Current driven by voltage source through resistive termination to ground plane. Dotted line: order-20 Arnoldi model with $s_0/(2\pi) = 0.5 + 2i$ GHz. Dash line: order-15 Arnoldi model with $s_0/(2\pi) = 0.5$ GHz. Solid line: response obtained without model reduction, as well as the piecewise-rational approximation obtained from combining models at separate expansion points; the results overlap. Note that the expansion at $0.5 + 2i$ Ghz gives a very good approximation up to 3 GHz but is not good near $s = 0$ whearas the opposite is true of the expansion at 0.5 GHz.

no information about the frequency response can be obtained. CFH, using multiple expansions, will require too many expansions due to the limitation, once again, of a single-point explicit moment-matching.

Figure 9-1(b) demonstrates that the Arnoldi-based model-order reduction applied to the sparse model generated by the precorrected-FFT algorithm generates a rational approximant for all the currents that is sufficently accurate for radiated field calculations. The maximum amplitude of the radiated field was calculated at 3m from the parallel strips. The reduced-order model of the dense system achieves essentially an exact match to the actual field amplitude. The reduced-order model of the precorrected-FFT method is only slightly less accurate.

The second example is a large multiconductor interconnect structure, which illustrates the capability of the combined multilevel/model-order-reduction algorithm to analyze very large, complex problems. Several signal traces, 200 μm wide run 3mm over a 10cm wide ground plane, as shown in Figure 9-2. A narrow mesh of traces is located over the signal paths, 6mm above the ground plane.

An automatic discretization routine requires that the ground plane contain an "image" discretization of the narrow traces, resulting in a final discretization with a large number of unknowns. The full model contains 5270 capacitive nodes and 9161 inductor currents. The final matrix which must be factored has 9129 unknowns, containing 8.3×10^7 entries. The code

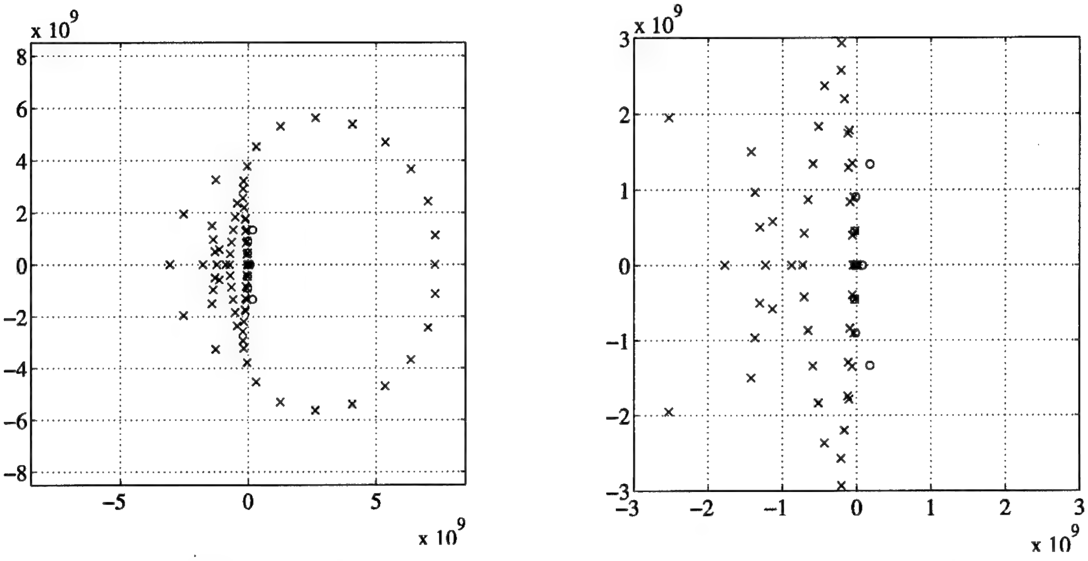


FIGURE 9-3: (x) Poles calculated by Arnoldi method. (*) poles calculated by moment-matching. The right hand plot is an expanded view of the poles near the origin.

using the full dense model would need about 3 gigabyte of storage, with 1.3 gigabytes needed just to store and factor the matrix. On a machine capable of 100MFLOPS sustained, about 5.6 hours would be needed for a single LU-factorization.

In contrast, the sparse model generated by the precorrected-FFT algorithm has only about 6.2×10^6 non-zero entries, corresponding to a factor of 13 reduction in model size. The code requires 600 megabytes of storage (a considerable portion of which is for storage of vectors needed for the recycled Krylov-subspace iterative matrix solution technique), and executes on a machine with 512 megabytes of physical memory without swap activity in the main solution phase. Figure 9-2 shows the response of the computed reduced-order model. It was possible to compute the order 20 model, which nearly spans the entire frequency range, in about 12 hours on an IBM RS6000/560 architecture.

9.5 Obtaining time-domain models

For many applications, it is sufficient to obtain the frequency-reponse of the system. Others, however, require time-domain information. At worst, once the frequency-domain information is available, a time-domain model can be quickly obtained by using the methods of [71], for example. A more elegant approach, however, would be to use the information available from the Arnoldi procedure to construct the time domain response, as generally once a rational approximant to the frequency response is available, the time-domain response is easily computed (see Eq. 9.15 below).

The first difficulty in obtaining time-domain models is that the rational approximation to the frequency response may contain poles that lie in the right-half plane. The corresponding time-domain models are thus not time-stable. This is a well-known problem for the methods that create Padé approximants by moment matching. The problem appears to be more pronounced for the Arnoldi approximants deriving from the series expansion of delay potentials. Figure 9-3 shows the calculated poles of the 85th order model of the two 30cm strips previously considered in Figure 9-1. Most of the poles of the Padé approximant lie in the left half plane; only a few stray poles are in the right half plane. The Arnoldi procedure in contrast generates poles in a cardoid-like pattern which extends well into the right-half-plane. What appears to be occurring is that in order to approximate the essential singularity at infinity of the exponential function, as the order of the approximation is increased, more and more poles are placed on a circular or oblong arc encompassing the origin, whose radius increases with the order of the approximation. In general we have observed that approximants generated by a non-symmetric Lanczos procedure (equivalent to a Padé approximation) generate fewer unstable poles than the Arnoldi procedure, for the type of series expansion of delay potentials considered in this paper.

The ideal resolution to this situation would be to develop a multi-point Krylov approximant which generated guaranteed-stable models. It is believed that the source of the unstable approximants is the expansion of the exponential delay factor in power series, so development of such an algorithm may require an alternative realization of the delay terms. A workable, if not ideal approach, however, is available which is similar to the CFH [31] algorithm.

First, for each pole in the frequency domain of interest, the residual of the eigenvalue/eigenvector pairs generated by the Arnoldi method is examined. If the residual is sufficiently small, and the pole is in the left half plane, it is retained for the final time-domain model. This procedure gives a set of poles from which a rational function can be constructed. It is then required that the frequency response of the final model and the actual frequency response match at a number of discrete points on the imaginary axis equal to the number of poles. This gives a linear system of equations that can be solved to obtain the residuals. This procedure at the same time gives the multi-point approximant, as the frequency data needed to calculate the final residuals can be taken from the nearest, and most accurate, frequency expansion points. Adequate results are obtained by choosing the frequency collocation points to be equal to the imaginary parts of the retained poles. Figure 9-4 shows the results of such a procedure.

A set of 30cm long strips was considered, with width 1mm and separation 1mm. Because of the smaller transverse dimensions, this model has a strong frequency response up to much higher frequencies than the previous strip model, which had 1cm transverse dimensions. Figures 9-4 (c)-(f) show the amplitude of the frequency response calculated at several different expansion points. Each response shown is derived from an order-50 model. The four models are patched together to obtain the final frequency response, shown as the solid lines in Figures 9-4(a) and (b). After deleting the unstable poles and recalulating the residues, the final model is obtained, whose frequency response is shown as the (+) in 9-4(a) and (b).

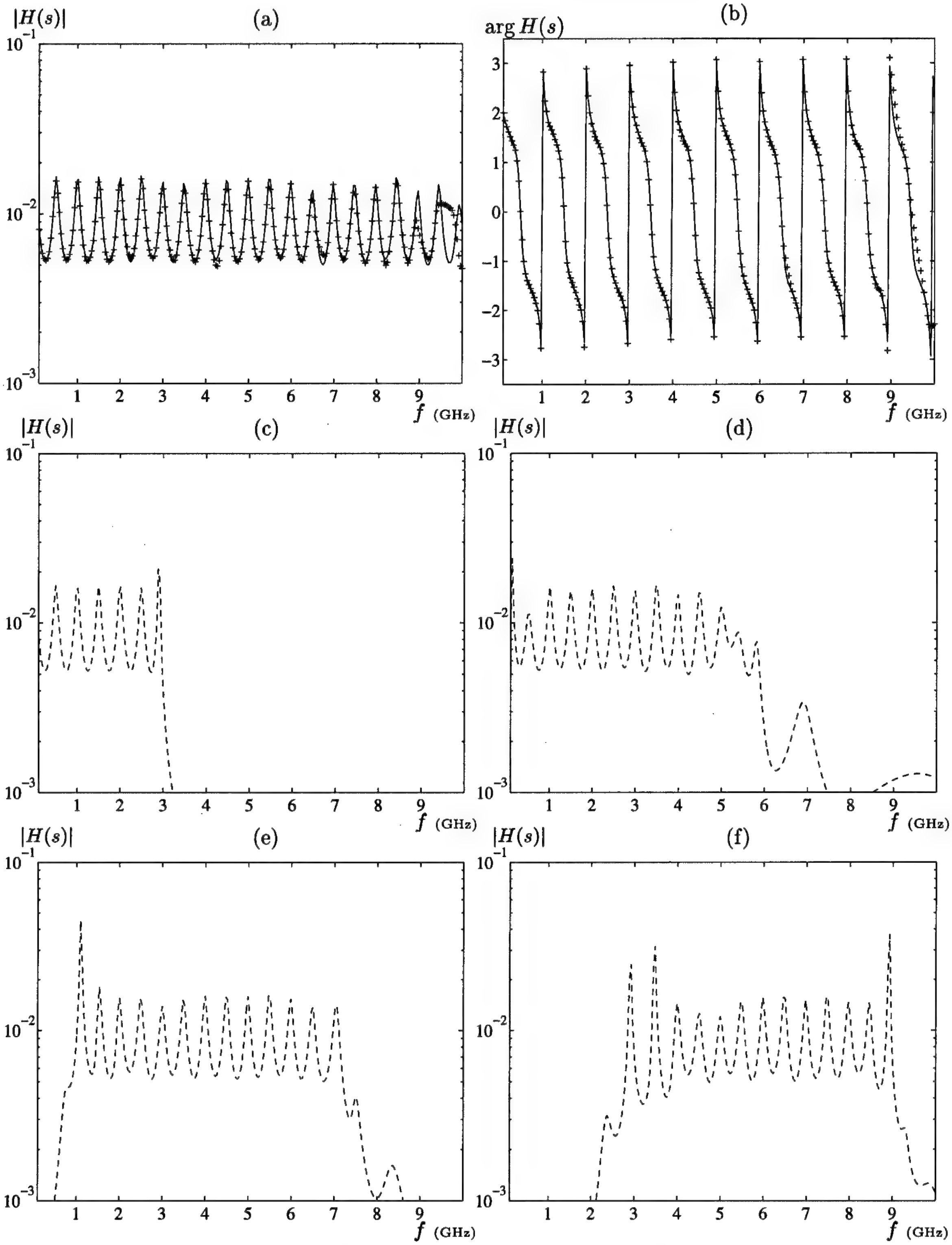


FIGURE 9-4: Frequency response functions for 30-cm long strips with 10-Ω termination driven by voltage source with 50-Ω internal impendance. (a) Magnitude of $H(s) = I(s)/V(s)$. Solid line shows overlapped multipoint frequency response, (+) show frequency response of final model after deletion of unstable poles and recalculation of residues. (b) phase of $H(s)$, same figure key as for (a). (c) Amplitude of $H(s)$, expansion point $s_0 = 0.5 + 0i$ Ghz (d) Amplitude of $H(s)$, expansion point $s_0 = 0.5 + 2i$ Ghz (e) Amplitude of $H(s)$, expansion point $s_0 = 0.5 + 4i$ Ghz (f) Amplitude of $H(s)$, expansion point $s_0 = 0.5 + 6i$ Ghz

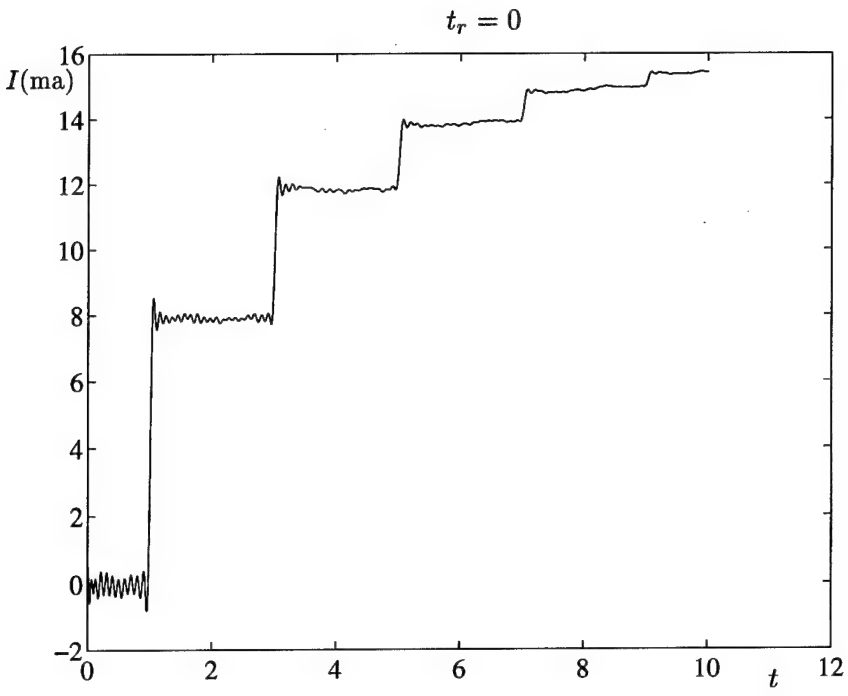


FIGURE 9-5: Time-domain step response of 30-cm long strips.

Once the frequency response has been constructed as the rational function $h(s)$,

$$h(s) = \sum_i \frac{r_i}{s - p_i} \tag{9.15}$$

it is an easy matter to obtain the time-domain response

$$h(t) = \sum_i r_i e^{p_i t}. \tag{9.16}$$

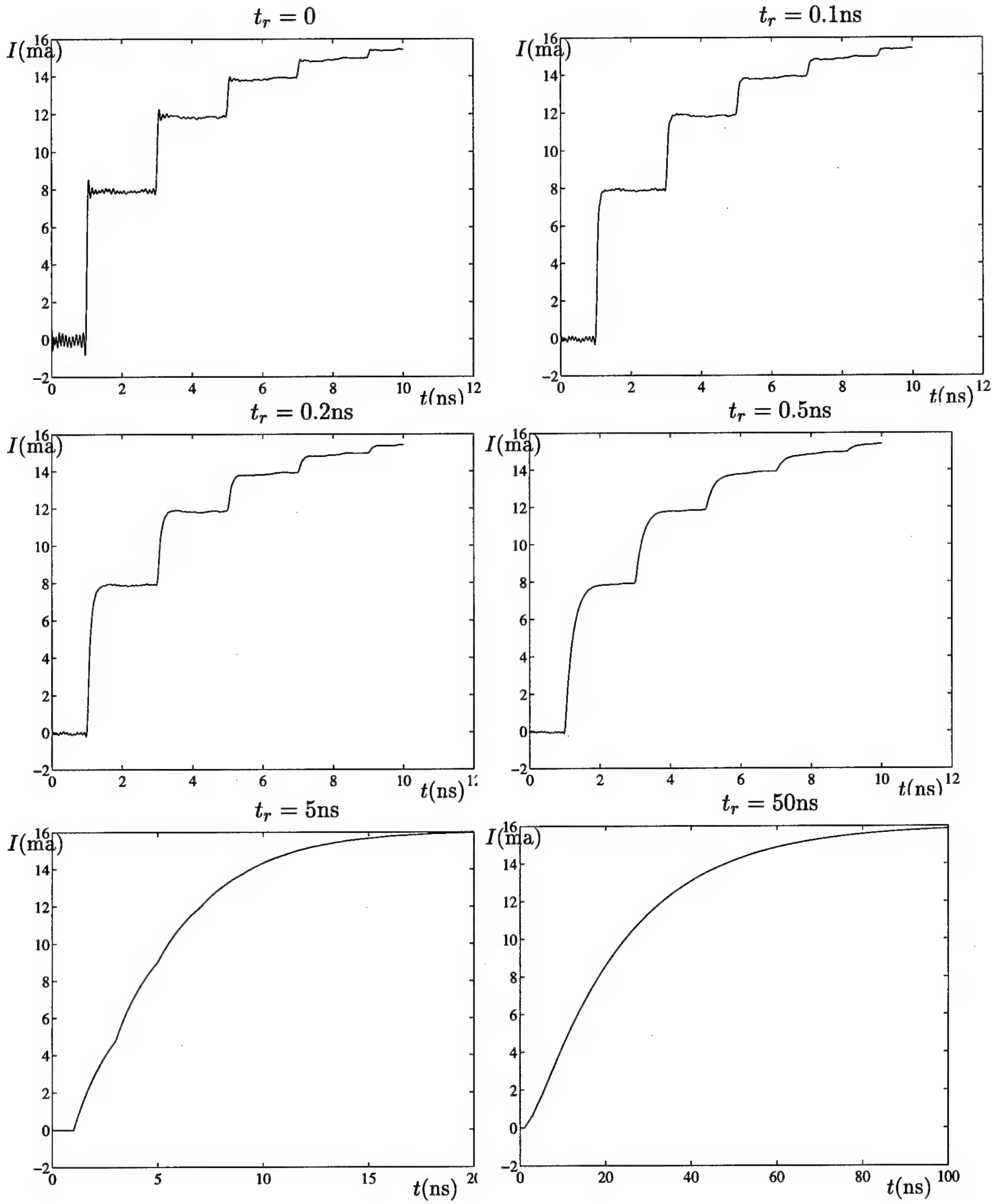
If the input $x(t)$ has the form of a rising exponential,

$$x(t) = \left[1 - e^{-\alpha t}\right] u(t) \tag{9.17}$$

where $u(t)$ is the unit step function, the output $y(t) = x(t) * h(t)$ is likewise easy to construct. Such a form for the input allows us to examine the effects of finite-risetime signals on the accuracy of the time-domain model. (For such an input, the risetime is about $2.3/\alpha$). The larger the risetime, the less accurate the frequency domain model needs to be to accurately calculate the output current, $y(t)$.

Figure 9-5 shows the time-domain response of the narrow-width strip model to a unit step input with zero risetime. This figure demonstrates several aspects of the combined algorithm.

The model is able to give a good qualitative representation of the dynamic response of a structure where the delay effects are very important. At least four reflections, representing nine one-way traversals of the strip by the wavefront, are represented by this single rational



$t_r = 0$

$t_r = 0.1$ns

$t_r = 0.2$ns

$t_r = 0.5$ns

$t_r = 5$ns

$t_r = 50$ns

FIGURE 9-6: Time-domain step response of 30-cm long strips with 10-Ω termination, varying input rise times t_r.

approximation. However, there are pronounced spurious oscillations in the reponse, especially for small times. In part this is due to the fact that we are examining higher frequencies than those for which the model is accurate, as this model is only accurate up to around 9 Ghz, but the frequency response is significant past that point.

If, however, the input signal has finite risetime, the truncation of higher-order frequency effects will not be visible in the final output current. The effect of varying the risetime is demonstrated in Figure 9-6. As the risetime is increased, the amplitude of the spurious oscillations becomes less pronounced. From these results, we conclude that certainly the model is likely to be accurate for frequencies up to 8 Ghz or so. Eventually the risetime becomes long enough that the output current simply follows the input – at this point the structure is behaving as a 60-Ω resistor.

9.6 Difficulties of the present approach

The integration of the model reduction procedure and the precorrected-FFT method exposes several shortcomings of both.

The application of the model reduction algorithm to the precorrected-FFT generated sparse model is easier to describe and implement if the projection and interpolation operators are frequency-independent, such as the polynomial-based operators used in this chapter. These operators, are, however, not the most accurate available. It is possible to adapt the algorithm to use the more accurate projection operators obtained from grid-collocation (for the far-field collocation operators, the frequency expansion points must be taken away from the origin) at a cost of a more complicated algorithm. It is unknown what effect such extension would have on the required computational resources.

However, one advantage of using polynomial interpolation is that it is possible to eliminate the precorrection step by modifying the kernels used for the grid- and short-range calculations[6]. The modification involves using an analytically "smoothed" kernel for the grid-potential calculation. The difference between the "smooth" kernel and the actual kernel can be incorporated into the direct potential calculation without reference to the grid since, for polynomial projection/interpolation operators, analytic forms are available for the "smooth" kernel. Such forms do not appear to be available for the grid-collocation operators.

When local corrections must be performed for every matrix vector product, this optimization should result in a significant savings in CPU time. If there are n panels in the system the m nearest-neighbors are included in the grid calculation, the precorrection cost is around order p^3m^3n. Even for the lowest accuracy likely to be useful in practical applications, the constant factor can easily be several thousand. For example, consider $p = 3$ with only nearest-neighbors considered in the direct calculation. For each cell, to perform the precorrection, the potential of the 3^3 grid charges must be calculated at 7^3 grid points. Done directly, this requires nearly

10,000 complex multiplies. The operation could also be done using a 3D FFT, at a similar cost. So, the number of floating-point double-precision multiplications needed per panel to perform the precorrection is around 40,000 divided by the average number of panels per cell. Empirically we have observed that the precorrection step represents by far the dominant cost in the matrix-vector product if it must be performed for each matrix-vector multiply. This partly accounts for the marginal speedups obtained, relative to direct methods, for the full-wave EM code, when compared to the speedups possible for the scalar Helmholtz and Laplace solvers described in previous chapters.

The other reason for the relatively disappointing results have to do with the nature of the iterative solver. When an iterative solver is used, at either each frequency point or each order in the model reduction procedure, a new iterative solve must be commenced. It is difficult for the iterative solver to effectively reuse information from previous solves. Reuse of the Krylov subspace is possible in the model reduction procedure, and has been observed to provide moderate performance improvements, but generally no reuse of information is possible for solves at differing frequency points. In contrast, when direct factorization is used in the model reduction procedure, the cost at each order is small once the matrix has been factored for the first order approximation. Similarly, when multiple solves at differing frequency points are required, the factored matrix provides a cheap and effective preconditioner for an iterative solve if the frequency points of the solution point and the frequency points for the factored matrix are reasonably close.

Additionally, iterative solution procedures only have substantial advantages over direct methods when applied to well-conditioned matrices, or, equivalently, when effective preconditioners are available. The matrices generated by the particular rPEEC implementation used are empirically very ill-conditioned. Often several hundred GMRES iterations are required to converge to moderate tolerances. Increasing the strain on the iterative solver, the iterative solver must be converged to a relatively high tolerance in order for the model reduction procedure to attain high orders of approximation.

The cost of the Arnoldi-based model reduction procedure can also become quite large at high order. The memory needed to store the Arnoldi vectors, and the cost required for the back-orthogonalization, grows rapidly with order and generally limits the maximum order that can be achieved for large problems to 100 or so before the memory and time needed by the Arnoldi procedure exceeds that required by the precorrected-FFT portion of the code.

Finally, this chapter has dealt only with single-input, single-output models. Most practical problems of concern, however, involve multiport models. The various Krylov-subspace methods extend quite naturally to the multi-input, multi-output (MIMO) case. For a single frequency expansion point, extending the algorithm presented here to MIMO problems is fairly straightforward. When multiple frequency points are required for an accurate response, however, the situation becomes more difficult. Extensions of the multipoint approximation procedure discussed here are possible, but awkward. A true multipoint Krylov-subspace algorithm, with stable approximations to the delays, would be the ideal solution.

10

Conclusion

In this thesis, a new approach was presented to reducing the CPU time and memory required to perform electromagnetic analyses of complicated 3-D structures. The primary computational tool developed was the grid-based, "precorrected-FFT" algorithm.

In Chapter 2 the problem of approximating singular potential functions was discussed, and a grid-collocation method described for representing charge potentials by the potential of a discrete set of point charges. These results motivated the development and analysis of the precorrected-FFT approach in Chapters 3 and 4, where indications of the proposed algorithm's viability were given. The final approach, using the grid-collocation operators, was shown to be well suited to applications which require solutions of integral equations with fixed parameters, such as problems associated with electrostatic analysis of integrated circuit packaging, on-chip interconnect, and micro-electro-mechanical systems, as discussed in Chapters 5 and 6. Additionally, parameter-dependent problems such as the scalar Helmholtz problems of Chapter 7, which could have practical application in computational acoustics, or the full-wave electromagnetic problems of Chapter 8, can be efficiently treated when information at a limited set of parameter values (e.g., single frequency solutions) is desired. For such problems, for moderately inhomogeneous geometries and moderate accuracies, the precorrected-FFT approach was more efficient than the competing fast multipole algorithm, in many cases by a large factor.

For problems that require solutions over a wide range of frequencies, additional algorithmic innovations were necessary. In order to effectively utilize the sparse representation generated by the precorrected-FFT method to efficiently compute the frequency response, it was necessary to develop a numerically stable algorithm for model order reduction of systems with delay elements. This model-order reduction technique was applied directly to the sparse model generated by the precorrected-FFT algorithm. The overall algorithm has substantial qualitative and quantitative advantages over standard full-wave electromagnetic analysis techniques. Qualitatively, a continuous analytic closed-form (pole-residue) expression for the frequency response is obtained, instead of samples at discrete points, and a pole-residue model may be obtained that can be directly incorporated into time-domain simulation. Quantitatively, for even moderate sized problems, substantially less memory is required by the new algorithm, and the frequency response can be obtained over a range of frequencies in the same time as the response is obtained at a few discrete points by standard techniques.

In Chapters 8-9 it was demonstrated that the combined approach of incorporating a relatively simple multilevel integral transform with the model reduction procedure can dramatically reduce the size of interconnect models generated by complicated three-dimensional interconnect structures. Compared to commonly-used techniques, on a problem with 10,000 unknowns, the resulting algorithm has a fivefold storage advantage, and computes a complete, continuous frequency response in the same time as standard techniques require for solutions at a few discrete frequency points. More importantly, the size of the sparse model and cost of generating the system response from it should grow considerably less rapidly with problem size than for standard techniques.

In light of these results we can draw some conclusions about the relative advantages of the grid-based algorithms proposed herein. A major advantage of the precorrected-FFT method is that it is based on the FFT and local interpolation operators, rather than on spherical-harmonics based shifting operators as in the fast multipole method. Thus, a range of kernels can be treated in the method while still preserving the high order of accuracy of multipole-based representations. This was the major motivation for the development of the algorithm.

The most salient drawback of the algorithm is its poor performance on very inhomogeneous problems. The use of the FFT to evaluate the grid potentials is inefficient in this case, since most of the data in the transform is zero. A similar problem occurs for in the layered media problem, when the panels are distributed among many layers of greatly varying width. In principle, algorithms can be developed to evaluate the grid potentials in the strongly inhomogeneous case by applying the algorithm of Chapter 3 recursively to obtain a multigrid-like algorithm. Such an algorithm would ideally inherit the efficiency of the grid-based representation discussed in this paper, while at the same time achieving the near $O(n)$ complexity of the fast multipole methods. The extension is straightforward for smooth kernels and one-dimensional implementations experiments have been encouraging, although there is some concern that computation of local interactions of the first grid level could introduce inefficiencies similar to those encountered in Chapter 9 due to local corrections. For oscillatory kernels a more cumbersome approach, such as proposed in [6], must be followed.

However, what our investigations reveal is that, in the larger picture, the issue of inhomogeneity, while jarring to the more mathematically inclined thinker, is, from an engineering perspective, actually of secondary importance. The results presented in this thesis lead us to believe that the precorrected-FFT method is adequate for most geometries of interest. What is of more immediate importance is to be able to efficiently integrate whatever sparse matrix representation is chosen with algorithms that can obtain useful models, for example time-domain models or frequency plots, rather than just simple single-point linear solutions, which are generally useful.

In this sense the more difficult issue is how to efficiently obtain local corrections. The approach of explicitly subtracting the grid contributions from the nearby interactions is either too computationally expensive, or too memory intensive, to incorporate in the present model-reduction procedure.

Additionally, at high order, the local correction step becomes too expensive to use even when amortized over many linear system solutions, such as in capacitance extraction and similar applications. This limits the degree of accuracy that can be achieved by grid-based methods.

Bibliography

[1] R. F. Harrington. *Field Computation by Moment Methods.* MacMillan, New York, 1968.

[2] S. H. Crandall. *Engineering Analysis.* McGraw-Hill, New York, 1956.

[3] L. Collatz. *The Numerical Treatment of Differential Equations.* Springer-Verlag, New York, third edition, 1966.

[4] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel. *Boundary Element Techniques.* Springer-Verlag, Berlin, 1984.

[5] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* M.I.T. Press, Cambridge, Massachusetts, 1988.

[6] A. Brandt. Multilevel computations of integral transforms and particle interactions with oscillatory kernals. *Computer Physics Communications*, 65:24–38, 1991.

[7] K. Nabors and J. White. FASTCAP: A multipole accelerated 3-D capacitance extraction program. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(11):1447–1459, November 1991.

[8] A. E. Ruehli and P. A. Brennan. Efficient capacitance calculations for three-dimensional multiconductor systems. *IEEE Transactions on Microwave Theory and Techniques*, 21(2):76–82, February 1973.

[9] S. M. Rao, T. K. Sarkar, and R. F. Harrington. The electrostatic field of conducting bodies in multiple dielectric media. *IEEE Transactions on Microwave Theory and Techniques*, MTT-32(11):1441–1448, November 1984.

[10] D. Colton and R. Kress. *Integral Equations Methods in Scattering Theory.* Krieger, Malabar, Florida, 1992.

[11] Youcef Saad and Martin Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, July 1986.

[12] R. W. Hockney and J. W. Eastwood. *Computer simulation using particles.* Adam Hilger, New York, 1988.

[13] A. Brandt and A. A. Lubrecht. Multilevel matrix multiplication and fast solution of integral equations. *J. Comp. Phys.*, 90:348–370, 1990.

[14] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60(2):187–207, September 15, 1985.

[15] V. Jandhyala, E. Michielssen, and R. Mittra. Multipole-accelerated capacitance computation for 3-d structures in a stratified dielectric medium using a closed form Green's function. *Int. J. Microwave and Millimeter-Wave Computer-Aided Eng.*, 5(2):68–78, 1995.

[16] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White. Multipole accelerated preconditioned iterative methods for three-dimensional potential integral equations of the first kind. *SIAM J. Sci. Stat. Comp.*, 15:713–735, May 1994.

[17] V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *J. Comp. Phys.*, 86:414–439, 1990.

[18] V. Rokhlin. Diagonal forms of translation operators for the Helmholtz equation in three dimensions. *Applied and Computational Harmonic Analysis*, 1:82–93, 1993.

[19] F. X. Canning. Transformations that produce a sparse moment matrix. *Journal of Electromagnetic Waves and Applications*, (4):893–913, 1990.

[20] C.C. Lu and W. C. Chew. Fast algorithms for solving hybrid integral equations. *IEEE Proceedings-H*, 140(6):455–460, December 1993.

[21] C. R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Comp.*, 13(4):923–947, July 1992.

[22] E. O. Brigham. *The Fast Fourier Transform and its Applications.* Prentice-Hall, Englewood Cliffs, 1988.

[23] C. F. van Loan. *Computational Frameworks for the Fast Fourier Transform.* SIAM, Philadelphia, 1992.

[24] J. W. Cooley and J. W. Tukey. An algorithm for the machine computation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.

[25] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. *Communications on Pure and Applied Mathematics*, 44:141–183, 1991.

[26] Lawrence T. Pillage and Ronald A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. CAD*, 9(4):352–366, April 1990.

[27] William B. Gragg. Matrix interpretations and applications of the continued fraction algorithm. *Rocky Mountain Journal of Mathematics*, 4(2):213–225, 1974.

[28] Peter Feldmann and Roland W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. CAD*, 14:639–649, May 1995.

[29] K. Gallivan, E. Grimme, and P. van Dooren. Asymptotic waveform evaluation via a Lanczos method. *Appl. Math. Lett.*, 7(5):75–80, 1994.

[30] T. K. Tang and M. S. Nakhla. Analysis of high-speed VLSI interconnects using the asymptotic waveform evaluation technique. *IEEE Trans. CAD*, 11:341–352, March 1992.

[31] Eli Chiprout and Michel S. Nakhla. Analysis of interconnect networks using complex frequency hopping (CFH). *IEEE Trans. CAD*, 14:186–200, February 1995.

[32] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag, New York, 1980.

[33] J. D. Jackson. *Classical Electrodynamics.* John Wiley, New York, 1975.

[34] A. D. McLaren. Optimal numerical integration on a sphere. *Math. Comput.*, 17:361–383, 1963.

[35] J. D. Jackson. *Classical Electrodynamics.* John Wiley, New York, 1975.

[36] Roger A. Horn and Charles R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, 1985.

[37] L. Berman. Grid-multipole calculations. *SIAM J. Sci. Comp.*, 16(5):1082–1091, September 1995.

[38] J. R. Phillips. Error and complexity analysis for a collocation-grid-projection plus precorrected-FFT algorithm for solving potential integral equations with Laplace or Helmholtz kernels. In *Proceedings of the 1995 Copper Mountain Conference on Multigrid Methods*, April 1995.

[39] M. Abramowitz and I. Stegun. *Handbook of mathematical functions.* National Bureau of Standards, Washington, DC, 1964.

[40] A. Mayo. The fast solution of Poisson's and the biharmonic equations on irregular regions. *SIAM J. Numer. Anal.*, 21(2):285–299, April 1984.

[41] A. H. Stroud. *Approximate Calculation of Multiple Integrals.* Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[42] V. I. Lebedev. Values of the nodes and weights of ninth to seventeenth order Gauss-Markov quadrature formulae invariant under the octahedron group with inversion. *Zh. vychisl. Mat. mat. Fiz.*, 15:48–54, 1975.

[43] V. I. Lebedev. Quadratures on a sphere. *Zh. vychisl. Mat. mat. Fiz.*, 16:293–306, 1976.

[44] C. R. Anderson. A method of local corrections for computing the velocity field due to a distribution of vortex blobs. *J. Comp. Phys.*, 62:111–123, 1986.

[45] David T. Borup and Om P. Gandhi. Calculation of high-resolution SAR distributions in biological bodies using the FFT algorithm and conjugate gradient method. *IEEE Transactions on Microwave Theory and Techniques*, 33:417–419, 1985.

[46] Tapan K. Sarkar, Ercument Arvas, and Sadasiva M. Rao. Application of FFT and the conjugate gradient method for the solution of electromagnetic radiation from electrically large and small conducting bodies. *IEEE Transactions on Antennas and Propagation*, 34:635–640, 1986.

[47] J. R. Phillips, M. Kamon, and J. White. An FFT-based approach to including non-ideal ground planes in a fast 3D inductance extraction program. In *Proc. of the Custom Int. Circuits Conf.*, May 1993.

[48] S. D. Senturia, R. M. Harris, B. P. Johnson, S. Kim, K. Nabors, M. A. Shulman, and J. K. White. A computer-aided design system for microelectromechanical systems (MEMCAD). *IEEE Journal of Microelectromechanical Systems*, 1(1):3–13, March 1992.

[49] Y. L. Le Coz and R. B. Iverson. A stochastic algorithm for high speed capacitance extraction in integrated circuits. *Solid State Electronics*, 35(7):1005–1012, 1992.

[50] P. Dewilde and Z. Q. Ning. *Models for Large Integrated Circuits.* Kluwer Academic Publishers, Boston, 1990.

[51] A. J. van Genderen and N. P. van der Meijs. Hierarchical extraction of 3-D interconnect capacitances in large regular VLSI structures. In *Proceedings of the Int. Conf. on Computer-Aided Design*, November 1993.

[52] A. H. Zemanian, R. P. Tewarson, C. P. Ju, and J. F. Jen. Three-dimensional capacitance computations for VLSI/ULSI interconnections. *IEEE Transactions on Computer-Aided Design*, 8(12):1319–1326, December 1989.

[53] A. Seidl, H. Klose, M. Svoboda, J. Oberndorfer, and W. Rösner. CAPCAL—a 3-D capacitance solver for support of CAD systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7(5):549–556, May 1988.

[54] P. E. Cottrell and E. M. Buturla. VLSI wiring capacitance. *IBM Journal of Research and Development*, 29(3):277–287, May 1985.

[55] T. Chou and Z. J. Cendes. Capacitance calculation of IC packages using the finite element method and planes of symmetry. *IEEE Trans. Computer-Aided Design*, 13:1159–1166, September 1994.

[56] K. Nabors and J. White. Multipole-accelerated capacitance extraction algorithms for 3-d structures with multiple dielectrics. *IEEE Trans. on Circuits and Systems*, 39(11):946–954, November 1992.

[57] J. Tausch and J. White. Mesh refinement strategies for capacitance extraction based on residual errors. In *Proceedings IEEE 5th topical meeting on electrical performance of electronic packaging*, October 1996.

[58] J. Tausch and J. White. Boundary integral solution of Laplace's equation in multi-layered media with high permittivity ratios. unpublished.

[59] J. Tausch and J. White. Capacitance extraction of 3-D conductor systems in dielectric media with high permittivity ratios. unpublished.

[60] K. S. Oh, D. Kuznetsov, and J. E. Schutt-Aine. Capacitance calculations in a multilayered dielectric medium using closed-form spatial green's functions. *IEEE Transactions on Microwave Theory and Techniques*, 42(8):1443–1453, August 1994.

[61] A. Allgower and K. Georg. *Numerical Continuation Methods.* Springer-Verlag, New York, 1990.

[62] R. W. Hamming. *Numerical methods for scientists and engineers.* Dover, New York, 1973.

[63] Hansruedi Heeb and Albert E. Ruehli. Three-dimensional interconnect analysis using partial element equivalent circuits. *IEEE Transactions on Circuits and Systems I:Fundamental Theory and Applications*, 39(11):974–982, November 1992.

[64] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, December 1987.

[65] L. Miguel Silveira, Mattan Kamon, and Jacob K. White. Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-D interconnect structures. In *Proceedings of the European Design and Test Conference*, Paris, France, March 1995.

[66] Hansruedi Heeb. EMI simulation using retarded partial element equivalent circuits and asymptotic waveform evaluation. In *Proceedings of the 11th Zurich symposium and technical exhibition on electromagnetic compatibility*, pages 191–195, Zurich, Switzerland, March 1995.

[67] Y. Saad. On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Math. Comp.*, 48:651–662, 1987.

[68] H. A. van der Vorst. An iterative solution method for solving $f(a)x = b$, using Krylov subspace information obtained for the symmetric positive definite matrix a. *J. Comp. Appl. Math.*, 18:249–263, 1987.

[69] R. Telichevesky, K. Kundert, and J White. Efficient AC and noise analysis of two-tone RF circuits. In *Proceedings 33rd Design Automation Conference*, Las Vegas, Nevada, June 1996.

[70] K. Gallivan, E. Grimme, and P. van Dooren. Padé approximation of large-scale dynamic systems with Lanczos methods. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, Lake Buena Vista, FL, December 1994.

[71] L. Miguel Silveira, Ibrahim M. Elfadel, and Jacob K. White. Efficient Frequency-Domain Modeling and Circuit Simulation of Transmission Lines. *IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part B*, 17(11), November 1994.